CARIBBEAN EXAMINATIONS COUNCIL

CAPE® Computer Science

SYLLABUS
SPECIMEN PAPER
MARK SCHEME
SUBJECT REPORTS

Macmillan Education Ltd
4 Crinan Street, London, N1 9XW
A division of Springer Nature Limited
Companies and representatives throughout the world

www.macmillan-caribbean.com

ISBN 978-0-230-48244-9 AER

First published 2014
This revised edition published 2020

Designed by Macmillan Education Limited

Cover design by Macmillan Education Limited and Red Giraffe

CAPE® Computer Science Free Resources

LIST OF CONTENTS

Computer Science

Computer science is the study of the theoretical foundations of information and computation and their implementation and application in computer systems. The CAPE Computer Science Syllabus provides persons with advanced knowledge, skills and attitudes to enable them to understand the uses and the impact of computer technologies, and to use the technology to create new computer applications for all areas of human activity. This syllabus provides opportunity for the acquisition of knowledge, skills and attitudes as preparation for further studies in Computer Science and the world of work.

This syllabus consists of two Units, each comprising three Modules.

Unit 1: Fundamentals of Computer Science

Module 1 – Computer Architecture and Organisation
Module 2 – Problem-Solving with Computers
Module 3 – Programming

Unit 2: Further Topics in Computer Science

Module 1 – Data Structures
Module 2 – Software Engineering
Module 3 – Operating Systems and Computer Networks

CARIBBEAN EXAMINATIONS COUNCIL

Caribbean Advanced Proficiency Examination

CAPE®

COMPUTER SCIENCE SYLLABUS

Effective for examinations from May/June 2009

Published by the Caribbean Examinations Council

Correspondence related to the syllabus should be addressed to:

The Pro-Registrar
Caribbean Examinations Council
Caenwood Centre
37 Arnold Road, Kingston 5, Jamaica, W.I.

Telephone: (876) 630-5200
Facsimile Number: (876) 967-4972
E-mail address: cxcwzo@cxc.org
Website: www.cxc.org

Contents

This document CXC A19/U2/08 replaces CXC A19/U2/03 issued in 2003.

Please note that the syllabus was revised and amendments are indicated by italics.

Revised 2008

Please check the website, www.cxc.org for updates on CXC's syllabuses.

Introduction

The Caribbean Advanced Proficiency Examination (CAPE) is designed to provide certification of the academic, vocational and technical achievement of students in the Caribbean who, having completed a minimum of five years of secondary education, wish to further their studies. The examinations address the skills and knowledge acquired by students under a flexible and articulated system where subjects are organised in 1-Unit or 2-Unit courses with each Unit containing three Modules. Subjects examined under CAPE may be studied concurrently or singly.

The Caribbean Examinations Council offers three types of certification. The first is the award of a certificate showing each CAPE Unit completed. The second is the CAPE diploma, awarded to candidates who have satisfactorily completed at least six Units, including Caribbean Studies. The third is the CAPE Associate Degree, awarded for the satisfactory completion of a prescribed cluster of seven CAPE Units including Caribbean Studies and Communication Studies. For the CAPE diploma and the CAPE Associate Degree, candidates must complete the cluster of required Units within a maximum period of five years.

Recognised educational institutions presenting candidates for CAPE Associate Degree in one of the nine categories must, on registering these candidates at the start of the qualifying year, have them confirm in the required form, the Associate Degree they wish to be awarded. Candidates will not be awarded any possible alternatives for which they did not apply.

Computer Science Syllabus

◆ RATIONALE

The widespread application of Computer Science, as embodied in the tools and techniques for gathering, manipulating, analysing and disseminating information, made possible because of dramatic improvements in computer and telecommunications technologies, has significantly changed society. A large proportion of business transactions is performed over computer networks. Multi-media computers have had a significant impact on the way in which people learn and on the way they seek entertainment. Moreover, the increased integration of computer and telecommunications technology, exemplified by the Internet and associated technologies, has led to an increased globalisation of the world economy. Computer Science, including the Internet, has significantly changed personal communication, commerce and the way in which academic research is conducted. Moreover, continuing developments in this field, *including the increased use of mobile networks and* the further improvement and decreasing cost of computer hardware, mean that the world has not seen the last of these changes.

The increasing importance of computer-based applications provides an important economic opportunity for the region. In recognition of this, a number of regional governments have made the provision of information services, including computer programming and software engineering, an important element in their economic development plans.

However, in order for the Caribbean to become an integral part of this new world and to take advantage of the economic opportunities it offers, citizens need to be able to use existing computer-based systems and to create and maintain them. The latter requires a solid foundation in Computer Science. Thus, Caribbean students need to acquire advanced knowledge, skills and attitudes to enable them to understand the uses and the impact of computer technologies, and to use the technology to create new computer applications for all areas of human activity. The syllabus is intended primarily for people who want to pursue a professional career in Computer Science or related disciplines and provides the opportunity for the acquisition of relevant knowledge, skills and attitudes as preparation for further studies in Computer Science and the world of work.

◆ AIMS

The syllabus aims to:

1. *develop a range of cognitive skills, including critical thinking skills;*

2. *develop an understanding of the components, the architecture and the organisation of a computer system;*

3. *equip students with the knowledge necessary to make informed decisions about the selection of components of computer systems;*

4. *develop an understanding of the problem-solving process;*

5. *equip students with skills to create algorithms to solve problems;*

6. *develop skills to write correct programs to solve problems;*

7. *develop an understanding of the concepts of software engineering;*

8. *provide students with an understanding of abstract data types and their usefulness for manipulating data;*

9. *develop skills in using essential tools and techniques in system development;*

10. *develop an appreciation for the characteristics of operating systems and their applications;*

11. *develop an understanding of how computer networks can be used to connect computers together, regardless of distance;*

12. *equip students with skills to design simple computer networks.*

◆ SKILLS AND ABILITIES TO BE ASSESSED

The skills that students are expected to have developed on completion of this syllabus have been grouped under three headings:

(i) Knowledge and Comprehension;
(ii) Application and Analysis;
(iii) Synthesis and Evaluation.

Knowledge and Comprehension

The ability to:

- identify, recall, and grasp the meaning of basic facts, concepts and principles;
- select appropriate ideas, match, compare and cite examples of facts, concepts, and principles in familiar situations;
- explain phenomena in terms of generally applicable principles.

Application and Analysis

The ability to:

- use facts, concepts, principles and procedures in unfamiliar situations;
- transform data accurately and appropriately and use common characteristics as a basis for classification;
- identify and recognise the component parts of a whole and interpret the relationships between those parts;
- identify causal factors and show how they interact with each other; infer, predict and draw conclusions;
- recognise the limitations and assumptions of data gathered in an attempt to solve a problem.

Synthesis and Evaluation

The ability to:

- make reasoned judgements and recommendations based on the value of ideas and information and their implications;
- use the computer and computer-based tools to solve problems;
- justify the appropriate application of techniques of problem-solving;
- select, justify and apply appropriate techniques and principles to develop data structures and application programs for the solution of a problem.

◆ PRE-REQUISITES OF THE SYLLABUS

Any person with a good grasp of the Caribbean Secondary Education Certificate (CSEC) Information Technology *or Mathematics* syllabuses, or their equivalent, should be able to pursue the course of study defined by this syllabus. However, successful participation in the course of study will also depend on the possession of good verbal and written communication skills.

◆ STRUCTURE OF THE SYLLABUS

This syllabus consists of two Units comprising three Modules each of 50 hours. Although the Units are independent of each other, together they provide a comprehensive introduction to the field of Computer Science.

UNIT 1: FUNDAMENTALS OF COMPUTER SCIENCE

Module 1 - COMPUTER ARCHITECTURE AND ORGANISATION
Module 2 - PROBLEM-SOLVING WITH COMPUTERS
Module 3 - PROGRAMMING

UNIT 2: FURTHER TOPICS IN COMPUTER SCIENCE

Module 1 - DATA STRUCTURES
Module 2 - SOFTWARE ENGINEERING
Module 3 - OPERATING SYSTEMS AND COMPUTER NETWORKS

◆ UNIT 1: FUNDAMENTALS OF COMPUTER SCIENCE

MODULE 1: *COMPUTER ARCHITECTURE AND ORGANISATION*

GENERAL OBJECTIVES

On completion of this Module, students should:

1. understand *the workings of* the components of computer-based systems;

2. *develop an appreciation of the functional components of the computer system, including the characteristics, performance and interactions.*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
1. *state the purpose of the main components of a computer system;*	*Hardware Components* *Input/output devices: port connectivity; speed; quality of output; specialised devices.* *Memory types: ROM; RAM; EPROM; EEPROM.* *Memory features: speed; size; word size; volatility.* *Storage devices: capacity, access speed, access method, portability.* *Security: surge protectors, voltage regulators, Uninterruptible Power Supplies (UPS).* *Types of computers: supercomputer, mainframe, microcomputer, Laptop, PDA.*
2. *describe the basic building blocks of a computer;*	*Computer Architecture* *Truth tables (refer to symbols on page 36).* *Logic gates; Flip flops; registers; counters; multiplexors; encoders, decoders.*
3. *explain how data is represented in a computer system;*	*Data Representation* *Bits; bytes; fixed (signed magnitude, ones and twos complement) and floating point (sign, mantissa and exponent) numbers and character representation; number bases.*

UNIT 1
MODULE 1: *COMPUTER ARCHITECTURE AND ORGANISATION (cont'd)*

SPECIFIC OBJECTIVES	**CONTENT**
Students should be able to:	
4. *describe the main characteristics of a processor.*	*Computer Organisation* *CPU components (ALU, CU, Registers), instruction format (addresses per instruction, fixed length vs variable length), types (data manipulation control and input/output) and sets; instruction fetch, decode and execute.* *Clock speed, cache memory.*

Suggested Teaching and Learning Activities

To facilitate students' attainment of the objectives of this Module, teachers are advised to engage students in the teaching and learning activities listed below.

1. Site visits to computer sales companies to view the various components of a computer system.

2. Invite computer professionals to talk to students on topics relating to the components of a computer system.

3. Divide class into groups and each group asked to conduct research on a topic related to the components of a computer system. Each group will then be required to present a report to the class. Students should be encouraged to gather updated information from various sources such as the Internet, current computer magazines, books and by interviewing computer professionals.

4. View interactive video tapes and Compact Disc, together with training materials on the components of a computer system.

5. Provide students with opportunities to talk to the class on a topic relating to the components of a computer system. Teacher will assign topics to students.

UNIT 1
MODULE 1: *COMPUTER ARCHITECTURE AND ORGANISATION (cont'd)*

RESOURCES

Bradley, R.	*Understanding Computer Science for Advanced Level,* London: Stanley Thornes, 2005.
Heathcote, P.	*A Level Computing,* London: Letts, 2005.
Parsons, J. and Oja, D.	*Computer Concepts,* Albany, New York: International Thomson Publishing Company, 2004.
Shelly, G., Cashman, T. and Vermaat, M.	*Discovering Computers 2008, Boston:* Thomson Course Technology, 2008.

UNIT 1
MODULE 2: *PROBLEM-SOLVING WITH COMPUTERS*

GENERAL OBJECTIVES

On completion of this Module, students should:

1. *understand the problem-solving process;*

2. *appreciate the role and importance of algorithms in the problem-solving process;*

3. *understand the process of developing algorithms.*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
1. *explain the concept of problem-solving;*	*Definition of problem-solving.*
2. *describe the stages in the problem-solving process;*	*Problem definition; problem analysis; identify and evaluate possible solutions; select and justify the optimal solutions; implementation and review.*
3. *explain the concept of an algorithm;*	*Definition; algorithm as a problem-solving strategy; its role and importance in the problem-solving process.*
4. *identify the necessary properties of algorithms that are well designed;*	*A general solution to the problem, clearly defined and unambiguous steps, finite number of steps, and flow of control from one process to another.*
5. *identify ways of representing algorithms;*	*Inclusion of narrative, flow charts and pseudocode.*
6. *explain constructs used in structured programming;*	*Input and output statements.* *Control Structures:* *Sequencing; Selection; Iteration or repetition (bounded, for example, fixed number of iterations and unbounded, for example, sentinel control); Assignment statement.*
7. *interpret algorithms from case problems;*	*Determination of output and correctness of a given algorithm (the algorithm may be expressed in narrative, flow charts or pseudocode).*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
8. *correct algorithms from case problems;*	*Determination of whether an algorithm achieves its stated objective and if not provision of the correct algorithm.*
9. *develop algorithms from case problems;*	
10. *explain the need for developing the logic of a computer program.*	*Algorithms as logically sequenced instructions.*

Suggested Teaching and Learning Activities

To facilitate students' attainment of the objectives of this Module, teachers are advised to engage students in the teaching and learning activities listed below.

1. *Engage students in a discussion leading to the definition of a problem. The activity should be geared to the students communicating their perspective of a problem. The teacher should then give feedback on those perspectives by identifying problems in different scenarios.*

2. *Encourage students to have an appreciation that not every problem can be solved using the computer. From a list of problems, the students should distinguish between problems that can be solved by using a computer and those which cannot be solved using the computer.*

3. *Give a set of scenarios in which there are either opportunities or problems encountered by an entity such as an organisation or a school. Students are required to (a) identify a problem, (b) formulate a problem statement, (c) suggest two possible solutions, and (d) recommend one of the solutions and justify the choice.*

4. *Use algorithms to solve simple tasks, for example, to compute the sum of a set of numbers.*

5. *Use different program constructs in developing algorithms.*

RESOURCES

Bradley, R.	*Understanding Computer Science for Advanced Level,* London: Stanley Thornes, 2005.
Heathcote, P.	*A Level Computing,* London: Letts, 2005.

Kendall, K. and Kendall, J.	*Systems Analysis and Design,* New Jersey: Prentice Hall, 2007.
Parsons, J. and Oja, D.	*Computer Concepts,* Albany, New York: International Thomson Publishing Company, 2004.
Shelly, G., Ashman, T. and Vermaat, M.	*Discovering Computers 2008, Boston:* Thomson Course Technology, 2008.

UNIT 1
MODULE 3: *PROGRAMMING*

GENERAL OBJECTIVES

On completion of this Module, students should:

1. *appreciate the need for different programming languages and program translation;*

2. *develop the ability to implement solutions to problems using a programming language.*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
1. *identify the characteristics of different programming paradigms;*	*Procedural or Imperative, Object-oriented, Functional and Declarative and others (for example, Aspect and Scripting).*
2. *explain the need for different programming languages;*	*Appropriateness to application (web application, games, formula translation, application for mobile devices).*
3. *explain how assemblers, compilers, virtual machines and interpreters are involved in the execution of High-level programming languages;*	*Stages in the translation process: lexical analysis; syntax analysis; semantic analysis; intermediate code generation; code optimization; code generation.* *Role of preprocessors; linkers.*
4. *assign values to declared variables;*	*Declare variables using appropriate names. Use appropriate and primitive data types (integer, float, double, char and enumerated).*
5. *use input and output statements;*	*Input data into variables; output formatted data from variables; print headings.*
6. *choose appropriate conditional and iterative constructs;*	
7. *use conditional and iterative control constructs;*	
8. *use arrays in programs;*	*Read data into arrays, output data from arrays, manipulate or modify data in arrays. Character arrays (strings).*

UNIT 1
MODULE 3: *PROGRAMMING (cont'd)*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
9. *apply the techniques of structured decomposition to reorganise a program into smaller pieces;*	*Write simple functions; programs should be clear, orthogonal (small blocks of code) and simple.*
10. *implement algorithms to solve a given problem;*	*Write, test and debug programs; syntax and semantic errors; use of range tests and desk checks; code debugging strategies (trace tables, use of 'watches' to examine the values of variables).*
11. *use records as a means of grouping related information;*	*The concept of 'struct' in C.*
12. *use text files to store data and records;*	*File operations: open, close, read, write, append.*
13. *develop good programming style.*	*White space (proper spacing), indentation, appropriate comments.*

CANDIDATES WILL BE ASSESSED IN PROCEDURAL 'C' ONLY.

Suggested Teaching and Learning Activities

To facilitate students' attainment of the objectives of this Module, teachers are advised to engage students in the teaching and learning activities listed below.

1. *Critique previously written programs focusing, for example, on the use of structure, constructs, comments, indentation, variable names and error handling.*
2. *Divide class into groups and ask each group to conduct research on a topic related to the implementation of different data structures with respect to performance. Each group will then be required to present a report to the class.*
3. *Develop test cases to illustrate the importance of testing.*
4. *Divide students into groups to research different languages, paradigms and translators and to examine the weaknesses and strengths of each language, paradigm and translator.*

RESOURCES

Shelly, G., Ashman, T. and Vermaat, M. — *Discovering Computers 2008, Boston:* Thomson Course Technology, 2008.

◆ UNIT 2: FURTHER TOPICS IN COMPUTER SCIENCE

MODULE 1: *DATA STRUCTURES*

GENERAL OBJECTIVES

On completion of this Module, students should:

1. *appreciate the use of abstract data types (ADTs) in the efficient manipulation of data;*

2. *understand basic algorithms for sorting and searching.*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
1. *describe the concept of abstract data types (ADTs);*	
2. *distinguish among ADTs;*	*Stacks (LIFO), queues (FIFO), singly linked list (INSERT and DELETE): definition, structure and operation.*
3. *perform basic operations on standard ADTs using diagrams and algorithms;*	*Stacks: Push, Pop, Empty, Full.* *Queues: ENQUEUE, DEQUEUE.*
4. *implement basic ADTs using one-dimensional arrays;*	*Write programs to implement Stacks, Queues.*
5. *describe searching and sorting algorithms using one-dimensional arrays;*	*Linear search; binary search; simple selection sort; bubble sort.*
6. *implement searching and sorting algorithms.*	*Linear search; binary search; simple selection sort; bubble sort.*

Suggested Teaching and Learning Activities

To facilitate students' attainment of the objectives of this Module, teachers are advised to engage students in the teaching and learning activities listed below.

1. *Use scenarios to illustrate the application of Abstract Data Types.*

2. *Make reference to real-life situations that demonstrate ADTs in action; for example, adding and removing plates from a stack of plates; customers in a queue (line) in a bank.*

UNIT 2
MODULE 1: *DATA STRUCTURES* (cont'd)

RESOURCES

Heathcote, P.	*A Level Computing,* London: Letts, 2005.
Kendall, K. and Kendall, J.	*Systems Analysis and Design,* New Jersey: Prentice Hall Inc., 2005.
Shelly, G., Ashman, T. and Vermaat, M.	*Discovering Computers 2008, Boston:* Thomson Course Technology, 2008.
Sommerville, I.	*Software Engineering,* Harlow: Addison Wesley, 2006.

UNIT 2
MODULE 2: *SOFTWARE ENGINEERING*

GENERAL OBJECTIVES

On completion of this Module, students should:

1. *understand the phases of the software development life cycle;*

2. *have an appreciation for the methods, processes, tools and techniques used in software engineering.*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
1. explain the reasons for a structured approach to the software development process;	Increased dependence of many organisations on their computer systems.
	Crises in previous developments: for example, increasing costs of software development; dissatisfaction of users and management with the quality and suitability of software; increasing length and complexity of the software.
	Requirements for standard interfaces, both to users and to other software.
	Need for tighter control and management of process; visibility of the process; risk management.
	Importance of the need for the involvement of end users and management.
2. explain the attributes of a well-engineered software product;	Properties of well-engineered software: maintainability; dependability; efficiency; usability; portability; availability of appropriate documentation.
3. identify different generic software process models and examine their strengths and weaknesses;	Phases of the Software Development Life Cycle.
	Life Cycle Models: waterfall approach; evolutionary development including rapid prototyping; fountain approach; formal transformation; reuse-oriented approach.

SPECIFIC OBJECTIVES	**CONTENT**
Students should be able to:	
4. *outline* the main activities, tools, techniques and deliverables of the analysis phase;	*Requirements and Specification Process: feasibility study; requirements analysis.* *Tools and Techniques: Interviews, questionnaires, observations, review internal documents, prototyping, Data Flow Models (Data Flow Diagrams) and their use to document the flow of information: use of symbols to depict data stores, process, data flows and external entities; Data Dictionaries; Semantic Data Models (Entity-Relationship Diagrams), Object Models;* *Computer Aided Software Engineering (CASE) tools.* *Deliverables: requirements specification (feasibility report, functional and non-functional specification).*
5. *outline the main activities, tools, techniques and deliverables of the design phase;*	*Design process: architectural design; interface design; data structure design; algorithm design.* *Tools and techniques: Structure charts, HIPO chart, CASE tools.* *Design Methods: top-down, bottom-up, system structuring (sub-systems, modules, programs); Design Strategies: functional versus objected-oriented.* *Guidelines for screens, reports, user interfaces.* *Deliverables: system architecture, design specification.*
6. *outline the main activities, tools, techniques and deliverables of the implementation phase;*	*Coding process.*
7. *outline the main activities, tools, techniques and deliverables of the validation phase;*	*Need for the testing process, test plans; software inspection; software testing (unit inspection, acceptance test, test case design).*
8. *outline the main activities, tools, techniques and deliverables of the evolution phase.*	

Suggested Teaching and Learning Activities

To facilitate students' attainment of the objectives of this Module, teachers are advised to engage students in the teaching and learning activities listed below.

1. Identify organisations that use custom-built software applications. Students should be divided into groups and asked to interview both users and management of these organisations to determine the following:

   (i) methodology (Life Cycle model used);

   (ii) problems encountered during the development of the application(s);

   (iii) level of user involvement;

   (iv) lessons learned;

   (v) what steps could have been done differently and why;

   (vi) other relevant considerations.

   Students can present their findings to the class.

2. Divide students into groups to research various Life Cycle models, tools and techniques used during the analysis and design phases. Students should report on their findings, including the strengths and weaknesses of models, tools and techniques.

3. Invite professionals involved in developing software to make presentations to students to give them additional perspectives on issues relevant to the topics. The professionals should be encouraged to bring samples of deliverables.

4. Identify a 'problem' and engage students in developing a simple system which could solve the problem.

5. Present 'poorly-designed" screen layouts, data structures, reports and user interfaces and ask students to critique them, for example, focusing on the appropriate use of font type and size; colours, spacing, labelling or instructions, ease of use.

RESOURCES

Bradley, R.	*Understanding Computer Science for Advanced Level,* London: Stanley Thornes, 2005.
Heathcote, P.	*A Level Computing,* London: Letts, 2005.
Parsons, J. and Oja, D.	*Computer Concepts,* Albany, New York: International Thomson Publishing Company, 2004.
Shelly, G., Ashman, T. and Vermaat, M.	*Discovering Computers 2008, Boston:* Thomson Course Technology, 2008.
Sommerville, I.	*Software Engineering*, Essex: Pearson Educational Limited, 2006.

UNIT 2
MODULE 3: OPERATING SYSTEMS AND COMPUTER NETWORKS

GENERAL OBJECTIVES

On completion of this Module, students should:

1. *understand the functions of operating systems;*

2. *develop an appreciation for networking technology and applications.*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
1. *explain the main functions of operating systems;*	*Resource manager; interface.*
2. *describe* how operating systems have evolved over time from primitive batch systems to sophisticated multi-user systems;	*History of operating system development.*
3. *describe* the functions of operating systems;	**Operating system functions:** *Bootstrap process* *Process Management* *Definition* *Process states: Running, Ready, Blocked.* *Definition* *How the interrupt mechanism works* *Types of interrupt:* - *interrupt generated by the running process* - *Input/Output Interrupt* - *External Interrupt* - *Restart Interrupt* *Deadlock (what is Deadlock)* *The process control block (process descriptor)* *Scheduling Algorithms Pre-emptive (Shortest-Job-First (SJF), round robin) and Non-pre-emptive (FCFS), Shortest-Job-First (SJF)* *(explain the concepts only)*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
	Memory Management *Virtual Memory, paging, thrashing*
	File Management *Directories/Folders, Files*
	Security (of files) *User IDs, Passwords, Lockwords, Access control list, file encryption, file compression* *Activity logs*
	Interface (user) *Types of interfaces: Menu, command prompt, GUI and the manipulation of the interface*
	Device Management *Device drivers* *Interrupt handling (PCB)* *Input/output control* *Peripheral control, polling Buffering, Spooling.*
	Networking *Network management (user accounts, access logs)* *Networking Protocols (TCP/IP)*
4. *distinguish among networked, client-server, and distributed;*	*Network Architecture: Ethernet, FDDI.* *Network topology: Star, Ring, Bus, Hybrid.* *Network devices: Modems, switches, routers, bridges, network interface cards (NIC), hubs.* *Transmission Media: wired (twisted pair, fiber-optics, coaxial); wireless (satellite, microwave)* *IEE1394 (Firewire) and cable connectors using diagrams.* *Protocol:* *Transmission Control Protocol/Internet Protocol (TCP/IP), File Transfer Protocol (FTP), Hypertext*

SPECIFIC OBJECTIVES	CONTENT
Students should be able to:	
	Transfer Protocol (HTTP); Hypertext Transfer Protocol Secure Sockets Layer (HTTPS); IEEE802.11a/b; IEEE802.16g; characteristics of Voice Over Internet Protocol; *Open System Interconnection (OSI) model.* *Access Methods for mobile networks: CDMA, TDMA, GSM, GPRS* *Networking consideration:* *cost, security, management, expandability, interconnectivity, wired vs wireless* *Network Configuration:* *Types: Multi-user; client server, centralised vs. distributed system, peer to peer.* *Network Security:* *Firewalls*
5. design simple networks.	*Use diagrams to design networks.*

Suggested Teaching and Learning Activity

To facilitate students' attainment of the objectives of this Module, teachers are advised to engage students in the teaching and learning activity below.

Divide class into groups and each group asked to conduct research on the functions of one type of operating system with respect to convenience, efficiency, and the ability to evolve. Each group will then be required to present a report to the class. Students should be encouraged to gather updated information from various sources such as the Internet, current computer magazines, books and by interviewing computer professionals.

UNIT 2
MODULE 3: OPERATING SYSTEMS AND COMPUTER NETWORKS (cont'd)

RESOURCES

Ritchie, C.	*Operating Systems Incorporating UNIX and Windows*, London: Letts Educational, 2003.
Shelly, G., Ashman, T. and Vermaat, M.	*Discovering Computers 2008, Boston:* Thomson Course Technology, 2008.

◆ OUTLINE OF ASSESSMENT

Each Unit of the syllabus will be assessed separately. The scheme of assessment for each Unit will be the same. Candidate's performance on each Unit will be reported as an overall grade and a grade on each Module of the Unit. The assessment will comprise two components, one external and one internal.

EXTERNAL ASSESSMENT **(80%)**

Paper 01 (1 hour 30 minutes)	*Forty-five multiple-choice items, fifteen (15) from each Module. Each item is worth 1 mark.*	(30%)
Paper 02 (2 hours 30 minutes)	Six questions, two from each Module. Candidates will be expected to answer all questions.	(50%)

INTERNAL ASSESSMENT **(20%)**

Paper 03A

The Internal Assessment for each unit is compulsory.

Unit 1: Fundamentals of Computer Science

Candidates are expected to choose a problem for which a software solution is appropriate and create an algorithm for the solution using sequencing, selection, assignments, iteration (bounded and unbounded). They should represent their algorithms using any combination of narrative, flow charts and pseudocode. Candidates are expected to implement the algorithm in C using arrays with no less than five functions and create a test plan.

Unit 2: Further Topics in Computer Science

Candidates are expected to choose a problem for which a software solution exists and then develop the software using software engineering techniques. In particular, they are expected to demonstrate the tools and techniques used in the analysis of the software to be developed. They are then expected to design, code, and test their software using appropriate techniques.

Paper 03B

Private candidates are required to write an Alternative Paper to the Internal Assessment Paper.

MODERATION OF INTERNAL ASSESSMENT

An Internal Assessment Record Sheet will be sent each year to schools submitting students for the examinations.

All Internal Assessment Record Sheets must be submitted to CXC by May 31 of each year of the examination. A sample of assignments will be requested by CXC for moderation purposes. These samples will be re-assessed by CXC Examiners who moderate the Internal Assessment. Teachers' marks may be adjusted as a result of moderation. The Examiners' comments will be sent to teachers.

Copies of the students' submissions must be retained by the school until three months after publication by CXC of the examination results.

ASSESSMENT DETAILS

External Assessment

Paper 01 and Paper 02

The external assessment consists of two written papers. They are externally set and externally assessed. Together they contribute 80% of the final mark.

Paper 01 (1 hour 30 minutes)

1. Composition of the Paper

The paper will consist of forty-five (45) multiple-choice items, fifteen (15) from each Module. All questions are compulsory and knowledge of the entire Syllabus is expected. The paper will assess the candidate's knowledge across the breadth of the Syllabus.

2. Mark Allocation

The paper is worth 45 marks, with each question being allocated 1 mark.

3. Question Type

Questions may be presented using diagrams, data, graphs, prose or other stimulus material.

Paper 02 (2 hours 30 minutes)

1. Composition of Paper

This paper consists of six questions, two from each Module, arranged in three sections. Candidates are required to do all questions in each section.

2. Mark Allocation

This paper is worth 150 marks, each question is worth 25 marks.

3. Question Type

Each question may present a situation related to a specific topic in the syllabus and consists of three or four sub-questions. The required responses to a sub-question may range in length.

4. Award of marks

Marks will be awarded for knowledge and comprehension, application and analysis and synthesis and evaluation.

Internal Assessment (20% of Total Assessment)

Internal Assessment is an integral part of student assessment in the course covered by this syllabus. It is intended to assist students in acquiring certain knowledge, skills and attitudes that are associated with the subject. The activities for the Internal Assessment are linked to the syllabus and should form part of the learning activities to enable the student to achieve the objectives of the syllabus.

During the course of study for the subject, students obtain marks for the competence they develop and demonstrate in undertaking their Internal Assessment assignments. These marks contribute to the final marks and grades that are awarded to students for their performance in the examination.

The guidelines provided in this syllabus for selecting appropriate tasks are intended to assist teachers and students in selecting assignments that are valid for the purpose of Internal Assessment. The guidelines provided for the assessment of these assignments are intended to assist teachers in awarding marks that are reliable estimates of the achievement of students in the Internal Assessment component of the course. In order to ensure that the scores awarded by the teachers are not out of line with the CXC standards, the Council undertakes the moderation of a sample of the Internal Assessment assignments marked by each teacher.

Internal Assessment provides an opportunity to individualise a part of the curriculum to meet the needs of students. It facilitates feedback to the student at various stages of the experience. This helps to build the self-confidence of students as they proceed with their studies. Internal Assessment also facilitates the development of critical skills and abilities emphasised by this CAPE subject and enhances the validity of the examination on which candidate performance is reported. Internal Assessment, therefore, makes a significant and unique contribution to both the development of relevant skills and the testing and rewarding of students for the development of those skills.

The Caribbean Examinations Council seeks to ensure that the Internal Assessment scores are valid and reliable estimates of accomplishment. The guidelines provided in this syllabus are intended to assist in doing so.

Each candidate's total Internal Assessment mark for any Unit should be divided in three and allocated to each Module equally.

Fractional marks should not be awarded. Wherever the Unit mark is not divisible by three, then

(a) when the remainder is 1 mark, it should be allocated to Module 1
(b) when the remainder is 2, one of the marks should be allocated to Module 2 and the other mark to Module 3.

Paper 03A

UNIT 1: Fundamentals of Computer Science

1. **The aims of the project are to:**

   (i) develop candidate's personal insights into the fundaments of Computer science;

   (ii) provide opportunities for all candidates to show, with confidence, that they have mastered the syllabus.

2. **Requirements**

   Each candidate is expected to choose a problem for which a software solution is appropriate and create algorithms for the solution using sequencing, selection, assignments, and iteration (bounded and unbounded). They should represent their algorithms using narrative format and either flow charts or pseudocode. Candidates are expected to implement their algorithms as C programs using arrays with no less than five functions and using at least one file. They must also create a test plan that is used to test their implemented programs for correctness.

3. **Integration of Project into the course**

   (i) The activities related to Project work should be integrated into the course so as to enable candidates to learn and practise the skills of undertaking a successful project.

   (ii) Some time in class should be allocated for general discussion of project work. For example, discussion of how data should be collected, how data should be analysed and how data should be presented.

   (iii) Class time should also be allocated for discussion between teacher and student, and student and student.

4. **Management of Project**

(i) **Planning**

An early start to planning project work is highly recommended and the schedule of the dates for submission should be developed by teachers and candidates.

(ii) **Length**

The length of the report of the project should be *between 1500 and 2000* words excluding diagrams, graphs, tables and bibliographies.

(iii) **Guidance**

Each candidate should know the requirements of the project and its assessment process.

Although candidates may consult with resource persons besides the teacher the candidates submission should be his or her own work.

Candidates are not expected to work on their own. The teacher is expected to give appropriate guidance at all stages of project work, for example, chapters to read, alternative procedures to follow and other sources of information.

(iv) **Authenticity**

Teachers are required to ensure that all projects are the candidates' work.

A recommended procedure is to:

(a) engage candidates in discussion;

(b) ask candidates to describe procedures used and summarise findings either orally or written;

(c) ask candidates to explain specific aspects of the analysis.

ASSESSMENT CRITERIA FOR THE PROJECT

General

It is recommended that candidates be provided with an assessment criteria before commencing the project.

(i) The following aspects of the project will be assessed:

  (a) Definition of problem;

  (b) Narrative and flow charts or pseudocode;

  (c) Coding of program;

  (d) Testing and presentation;

  (e) Communication of Information.

(ii) For each component, the aim is to find the level of achievement reached by the candidate.

(iii) For each component, only whole numbers should be awarded.

(iv) It is recommended that the assessment criteria be available to candidates at all times.

CRITERIA FOR MARKING INTERNAL ASSESSMENT PROJECT

The project will be graded out of a total of **60** marks and marks will be allocated to each task as outlined below. *Candidates will be awarded marks for communicating information in a logical way using correct grammar. These marks are awarded under Task 5.0 below.*

1.	**Definition of Problem**	**[4]**
	• Complete and accurate description of the problem	3-4
	• Partial description of the problem	1-2
2.	**Narrative and Flow Charts or Pseudocode**	**[15]**
	• Algorithms expressed in narrative format	(4)
	- Narrative is an accurate description of a solution	3-4
	- Narrative is a partially correct description of a solution	1-2
	• Algorithms expressed as flow charts or pseudocode	(6)
	- Flow chart/Pseudocode is logical, easy to follow and is an accurate description of a solution using the appropriate symbols or algorithmic structures	5-6
	- Flow chart/Pseudocode is organised, easy to follow for the most part, and is a clear description of a solution using the appropriate symbols or algorithmic structures	3-4
	- Flow chart/Pseudocode is not well organised, and description of solution lacks clarity	1-2
	• Demonstration of structured programming concepts	(5)
	- Program displays excellent use of structured programming concepts	5
	- Program displays good use of structured programming concepts	3-4
	- Program displays limited use of structured programming concepts	1-2
3.	**Coding of Program**	**[25]**
	• Structured decomposition using functions	(6)
	- Overall, program comprises functions as independent units	5-6
	- Program comprises most functions as independent units	3-4
	- Program comprises some functions as independent units	1-2
	• Use of appropriate data structures	(6)
	- Data structure chosen were appropriate for the problem definition	5-6
	- Data structure chosen were reasonable but not appropriate	3-4
	- Data structure chosen were inappropriate	1-2
	• Demonstration of the concept of structured programming	(3)
	- Evidence of looping structures	3
	- Evidence of conditional statements	2
	- Evidence of other structures (for example assignment, input, output)	1
	• Appropriate programming style and documentation	(4)
	- Appropriate document in significant areas	3-4
	- Standard indentation of code	1-2

- Evidence that code matches algorithm (4)
    - Code matches sequencing of algorithm 4
    - Code matches MOST of the sequencing of algorithm 3
    - Code matches SOME of the sequencing of algorithm 2
    - Sequencing of code inconsistent with algorithm 1
- Evidence of file manipulation (2)
    - Correct file types used, for example, text, binary, sequential, random 2
    - File used appropriately 1

4. **Testing and presentation** **[11]**
- Test Plan (3)
    - Test Plan with exhaustive data set 3
    - Test Plan with acceptable data set 2
    - Test Plan with minimal data set 1
- Test Results (5)
    - Normal input giving correct results 5
    - Extreme input giving correct results or appropriate error message 3-4
    - Erroneous input (for example, text when number required) giving appropriate error message 2
    - Incomplete input giving appropriate message 1
- Overall presentation (3)
    - Appropriate cover page 1
    - Use of table of contents 1
    - Sequencing in document easy to follow 1

5. **Communication of Information** **[5]**
- Communicates information in a logical way using correct grammar and appropriate jargon ALL of the time 4-5
- Communicates information in a logical way using correct grammar and appropriate jargon MOST of the time 2-3
- Communicates information in a logical way using correct grammar and appropriate jargon MOST of the time 1

TOTAL **60**

UNIT 2: **Further Topics in Computer Science**

1. **The aims of the project are to:**

(i) develop candidate's personal insights into further topics in Computer Science;

(ii) provide opportunities for all candidates to show, with confidence, that they have mastered the syllabus.

2. **Requirements**

Each candidate is expected to choose a problem for which a software solution exists and then develop the software using software engineering techniques. In particular, the candidate is expected to demonstrate appropriate choice of the tools and techniques used in the analysis of the software to be developed. They are then expected to design, code, and test their software using appropriate techniques.

The following are examples of the kinds of projects that students can develop for the Internal Assessment:

(i) *simple process scheduler for an operating system;*
(ii) *vehicle parking system to allocate spaces to vehicles in a parking lot;*
(iii) *system to manage a CD/DVD collection;*
(iv) *student registration system to keep track of student information, course grades and registration details.*

3. **Integration of Project into the course**

(i) The activities related to Project work should be integrated into the course so as to enable candidates to learn and practise the skills of undertaking a successful project.

(ii) Some time in class should be allocated for general discussion of project work. For example, discussion of how data should be collected, how data should be analysed and how data should be presented.

(iii) Class time should also be allocated for discussion between teacher and student, and student and student.

4. **Management of Project**

(i) **Planning**

An early start to planning project work is highly recommended and the schedule of the dates for submission should be developed by teachers and candidates.

(ii) **Length**

The length of the report of the project should be *between 1500 and 2000* words excluding diagrams, graphs, tables and bibliographies.

(iii) **Guidance**

Each candidate should know the requirements of the project and its assessment process.

Although candidates may consult with resource persons besides the teacher the candidates submission should be his or her own work.

Candidates are not expected to work on their own. The teacher is expected to give appropriate guidance at all stages of project work, for example, chapters to read, alternative procedures to follow and other sources of information.

(iv) **Authenticity**

Teachers are required to ensure that all projects are the candidates' work.

A recommended procedure is to:

(a) engage candidates in discussion;

(b) ask candidates to describe procedures used and summarise findings either orally or written;

(c) ask candidates to explain specific aspects of the analysis.

ASSESSMENT CRITERIA FOR THE PROJECT

General

It is recommended that candidates be provided with an assessment criteria before commencing the project.

(i) The following aspects of the project will be assessed:

(a) Specification of requirements;

(b) Design Specification;

(c) Coding and Testing;

(d) Communication of Information.

(v) For each component, the aim is to find the level of achievement reached by the candidate.

(vi) For each component, only whole numbers should be awarded.

(vii) It is recommended that the assessment criteria be available to candidates at all times.

CRITERIA FOR MARKING INTERNAL ASSESSMENT

Candidates will be awarded a total of 5 marks for communicating information in a logical way using correct grammar. The marks are awarded as shown in the mark scheme below.

1. Specification of requirements **[25]**

- Definition of problem (5)
  - Complete accurate description of the problem 5
  - Generally accurate description for the problem 4
  - Partially accurate description for the problem 3
  - Weak description for the problem 1-2
- Techniques of analysis used (5)
  - Sound and relevant techniques used 5
  - Mostly sound and relevant techniques 3-4
  - Techniques were partially sound and relevance was limited 1-2

- Use of Data Flow diagrams and E-R diagrams (9)

  Data Flow Diagrams (DFD) (3)

  Context Level
  - Complete and accurate diagram of all relevant entities, data flows. 3
  - Accurate diagram of most relevant entities, data flows. 2
  - Accurate diagram of few relevant entities, data flows. 1

- Level 1 Diagram (3)
  - Complete and accurate diagram of all relevant processes, data flows and major data stores 3
  - Accurate diagram of most relevant processes, data flows and major data stores 2
  - Accurate diagram of few relevant processes, data flows and major data stores 1

- Entity Relation Diagram (ERD) (3)
  - Complete and accurate diagram of all relevant entities and relationships. 3
  - Accurate diagram of most relevant entities and relationships.
  - Accurate diagram of few relevant entities and relationships 2

  1

- Functional and non-functional requirements (6)

  Functional Requirements (3)
  - Complete and accurate description of all requirements 3
  - Complete and accurate description of most requirements 2
  - Complete and accurate description of few requirements 1

- Non Functional Requirements (3)
  - Complete and accurate description of all requirements 3
  - Accurate description of most requirements 2
  - Accurate description of some requirements 1

2. **Design Specification** [14]

- System structuring (4)
  - Complete and accurate diagram of all processes 4
  - Accurate diagram of most processes 3
  - Accurate diagram of some processes 2
  - Accurate diagram of few processes 1

- User interface design (2)
  - Thorough analysis and appropriate justification of interface design 2
  - Partial analysis and justification of interface design 1

- Report design (2)
  - Appropriate and well implemented 2
  - Generally appropriate and satisfactorily implemented 1

- Algorithm design (3)
  - Appropriate and well implemented algorithm design 3
  - Generally appropriate algorithm design 2
  - General understanding of algorithm design 1

- Choice of appropriate data structures (3)
  - Appropriate and well implemented 3
  - Generally appropriate 2
  - Partially appropriate and implementation was limited 1

3. **Coding and Testing** [15]

- Code achieves functionality (5)
  - Code achieved functionality (documentation, error trapping, correct output, usability and reporting) 5
  - Code achieved some functionality (documentation, error trapping, correct output, usability and reporting) 3-4
  - Functionality was limited 1-2

- Code corresponds to design (5)
  - Code achieves all the design specifications 5
  - Code achieves most of the design specifications 3-4
  - Code achieves few of the design specifications 1-2

- Test plans (5)
  - Test Plan with exhaustive data set 5
  - Test Plan with acceptable data set 3-4
  - Test Plan with minimal data set 1-2

4.	**Communication and Presentation**	**[6]**
	• Communicates information in a logical way using correct grammar and appropriate jargon ALL of the time	5-6
	• Communicates information in a logical way using correct grammar and appropriate jargon MOST of the time	3-4
	• Communicates information in a logical way using correct grammar and appropriate jargon SOME of the time	1-2
	TOTAL	**60**

◆ REGULATIONS FOR PRIVATE CANDIDATES

Candidates who are registered privately will be required to sit Paper 01, Paper 02 and Paper 03B. Paper 03B will test the student's acquisition of the skills in the same areas of the syllabus identified for the internal assessment. Consequently, candidates are advised to undertake a project similar to the project that the school candidates would normally complete and submit for internal assessment. It should be noted that private candidates would not be required to submit a project document.

◆ REGULATIONS FOR RESIT CANDIDATES

Resit candidates must rewrite Papers 01 and 02 of the examination for the year in which they re-register. Resit candidates may elect not to repeat the Internal Assessment component provided they rewrite the examination no later than two years following their first attempt.

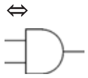
Resit candidates must be entered through a school, approved educational institution, or the Local Registrar's Office.

◆ ASSESSMENT GRID

The Assessment Grid for each Unit showing marks assigned to each paper and to Modules, and the percentage contribution of each paper to the total scores.

Papers	Module 1	Module 2	Module 3	Total	(%)
External Assessment Paper 01 Multiple Choice (1 hour 30 minutes)	(15)	(15)	(15)	(45)	(30)
Weighting	30	30	30	90	
Paper 02 Essay (2 hours 30 minutes)	50	50	50	150	(50)
Internal Assessment Paper 03A Paper 03B (1 hour 30 minutes)	20	20	20	60	(20)
Total	100	100	100	300	(100)

◆ LOGIC SYMBOLS

p, q, r	propositions
∧	conjunction
∨	(inclusive) disjunction
~	negation
→	conditionality
↔	bi-conditionality
·	implication
⇔	equivalence
	AND gate
	OR gate
	NOT gate
T, 1	true
F, 0	false

◆ GLOSSARY

WORD	DEFINITION/MEANING	NOTES
analyse	examine in detail	
annotate	add a brief note to a label	Simple phrase or a few words only.
apply	use knowledge/principles to solve problems	Make inferences/conclusions.
assess	present reasons for the importance of particular structures, relationships or processes	Compare the advantages and disadvantages or the merits and demerits of a particular structure, relationship or process.
calculate	arrive at the solution to a numerical problem	Steps should be shown; units must be included.
classify	divide into groups according to observable characteristics	
comment	state opinion or view with supporting reasons	
compare	state similarities and differences	An explanation of the significance of each similarity and difference stated may be required for comparisons which are other than structural.
construct	use a specific format to make and/or draw a graph, histogram, pie chart or other representation using data or material provided or drawn from practical investigations, build (for example, a model), draw scale diagram	Such representations should normally bear a title, appropriate headings and legend.
deduce	make a logical connection between two or more pieces of information; use data to arrive at a conclusion	
define	state concisely the meaning of a word or term	This should include the defining equation/formula where relevant.
demonstrate	show; direct attention to...	
derive	to deduce, determine or extract from data by a set of logical steps some relationship, formula or result	This relationship etc., may be general or specific.

WORD	DEFINITION/MEANING	NOTES
describe	provide detailed factual information of the appearance or arrangement of a specific structure or a sequence of a specific process	Description may be in words, drawings or diagrams or any appropriate combination. Drawings or diagrams should be annotated to show appropriate detail where necessary.
determine	find the value of a physical quantity	
design	plan and present with appropriate practical detail	Where hypotheses are stated or when tests are to be conducted, possible outcomes should be clearly stated and/or the way in which data will be analyzed and presented.
develop	expand or elaborate an idea or argument with supporting reasons	
diagram	simplified representation showing the relationship between components.	
differentiate/ distinguish (between/ among)	state or explain briefly those differences between or among items which can be used to define the items or place them into separate categories.	
discuss	present reasoned argument; consider points both for and against; explain the relative merits of a case	
draw	make a line representation from specimens or apparatus which shows an accurate relation between the parts	In the case of drawings from specimens, the magnification must always be stated.
estimate	make an approximate quantitative judgement	
evaluate	weigh evidence and make judgements based on given criteria	The use of logical supporting reasons for a particular point of view is more important than the view held; usually both sides of an argument should be considered.
explain	give reasons based on recall; account for	
find	locate a feature or obtain as from a graph	

WORD	DEFINITION/MEANING	NOTES
formulate	devise a hypothesis	
identify	name or point out specific components or features	
illustrate	show clearly by using appropriate examples or diagrams, sketches	
interpret	explain the meaning of	
justify	explain the correctness of	
investigate	use simple systematic procedures to observe, record data and draw logical conclusions	
label	add names to identify structures or parts indicated by pointers	
list	itemise without detail	
measure	take accurate quantitative readings using appropriate instruments	
name	give only the name of	No additional information is required.
note	write down observations	
observe	pay attention to details which characterise a specimen, reaction or change taking place; to examine and note scientifically	Observations may involve all the senses and/or extensions of them but would normally exclude the sense of taste.
outline	give basic steps only	
plan	prepare to conduct an investigation	
predict	use information provided to arrive at a likely conclusion or suggest a possible outcome	
record	write an accurate description of the full range of observations made during a given procedure	This includes the values for any variable being investigated; where appropriate, recorded data may be depicted in graphs, histograms or tables.

WORD	DEFINITION/MEANING	NOTES
relate	show connections between; explain how one set of facts or data depend on others or are determined by them	
sketch	make a simple freehand diagram showing relevant proportions and any important details	
state	provide factual information in concise terms outlining explanations	
suggest	offer an explanation deduced from information provided or previous knowledge. (... a hypothesis; provide a generalization which offers a likely explanation for a set of data or observations.)	No correct or incorrect solution is presumed but suggestions must be acceptable within the limits of scientific knowledge.
test	to find out, following set procedures	

Western Zone Office
2008/06/02

C A R I B B E A N E X A M I N A T I O N S C O U N C I L

ADVANCED PROFICIENCY EXAMINATION

SPECIMEN PAPER
MULTIPLE CHOICE QUESTIONS
FOR

COMPUTER SCIENCE - UNIT 1

Paper 01

READ THE FOLLOWING DIRECTIONS CAREFULLY

Each item in this test has four suggested answers lettered (A), (B), (C), (D). Read each item you are about to answer and decide which choice is best.

Sample Item

Which of the following is a function of process management in an operating system?

Sample Answer

(A) Repairing disk errors
(B) Scheduling
(C) Error reporting
(D) Partitioning

A ● C D

The best answer to this item is "Scheduling", so answer space (B) has been shaded.

There are 21 items in this specimen paper. However, the Paper 01 test consists of 45 items. You will have 120 minutes to answer them.

1. Which of the following descriptions BEST suits the function of Cache memory?

(A) It gives the CPU more rapid access to data.
(B) It increases data transfer rate between computer and printer.
(C) It speeds up access to data on the hard disk.
(D) It stores the operating system when the computer "boots".

2. Which of the following devices would help prevent data loss in the event of an electrical outage?

(A) Power strip
(B) Surge protector
(C) UPS
(D) Voltage regulator

3. A computer's word size is the

(A) length of an instruction
(B) maximum number of characters in a typed word
(C) storage capacity of the computer's memory
(D) number of bits that the CPU can manipulate at one time

4. The 8-bit two's complement representation of the decimal numeral -39 is:

(A) 00100110
(B) 10100101
(C) 11011001
(D) 11100110

5. The 8-bit Sign & Magnitude representation of the decimal numeral -25 is:

(A) 10011001
(B) 10100101
(C) 11100111
(D) 11100110

6. What is the purpose of the program counter in a microcomputer?

(A) To determine how many programs can be opened at one time
(B) To determine the sequence in which the program instructions are to be executed.
(C) To hold the number of the last instruction executed
(D) To keep a count of the number of instructions in memory

7. Which of the followingis NOT true of a flip-flop?

(A) It has two inputs.
(B) It has two outputs.
(C) Is a bistable device.
(D) It can function as a 1-bit memory.

8. The list I - VI represent steps in the computer-based problem solving process.

I. Analyze the problem
II. Code the solution
III. Define the problem
IV. Develop an algorithm
V. Maintain the program
VI. Test and debug the program

Which of the following represents the correct sequence of steps in the computer-based problem-solving process?

(A) I, II, III, IV, V, VI
(B) III, I, IV, II, VI, V
(C) III, II, VI, IV, V, I
(D) III, I, VI, II, V, VI

GO ON TO THE NEXT PAGE

Items **9 - 10** refer to the following algorithm.

```
Num = 5
For I = 1 to Num do
        Print I *2
EndFor
```

9. This algorithm is an example of

(A) unbounded iteration
(B) recursion
(C) selection
(D) bounded iteration

10. What is the output of the algorithm?

(A) 1 1
(B) 1 2 3 4 5
(C) 2 4 6 8 10
(D) I*2 I*2 I*2 I*2 I*2

11. What is the technical term for the graphical representation of an algorithm?

(A) Flowchart
(B) HIPO chart
(C) Narrative
(D) Pseudocode

12. Which of the following are properties of a well-designed algorithm?

I. A finite number of steps
II. Flow of control from one process to the next
III Correct syntax
IV. Ambiguous instructions

(A) I and II only
(B) I and III only
(C) I, II and III only
(D) II, III and IV only

13. Which of the following statements are true about programs and algorithms?

I. Every program uses algorithms.
II. All programs are algorithms.
III. Algorithms are derived from programs.

(A) I and II only
(B) I and III only
(C) II and III only
(D) I, II and III

14. What are the values of p and r after execution of the following algorithm?

```
p = 8
q = 4
r = 8

if (p > q) AND (q > r)
        p = + 1
else
        r = r - 1
end if
```

(A) p = 7, r = 8
(B) p = 9, r = 7
(C) p = 7, r = 7
(D) p = 8, r = 7

15. What is the error in the following C program which is intended to print the string constant ‘C programming is fun’?

```
#include<stdio.h>
{printf(“C programming is fun”); }
```

(A) The brackets (and) should be on different lines.
(B) The function main is missing.
(C) The braces { and } should be on different lines.
(D) Single quotation marks ‘ and ‘ should be used.

16. Programming languages may be classified by Generation and Paradigm among other things. Which of the following is NOT an example of a programming language paradigm?

(A) Imperative
(B) Functional
(C) Prerogative
(D) Declarative

Item **17** refers to the following algorithm.

```
#include<stdio.h>

main ( )
{
        int a, b, sum;
        a = 10;
        b = 20;
        sum = a + b;
        printf("The sum of the numbers is
        %d/n", sum);
 }
```

17. The program line: sum = a + b is an example of

(A) a function call
(B) a C operator
(C) a variable declaration
(D) the C assignment

18. What output would you expect from the following program?

```
#include<stdio.h>

main ( )
{
        printf("One——");
        printf("Two——");
        printf("Three\n");
 }
```

(A) One——Two——Three
(B) One——, Two——, Three
(C) One——
Two——
Three\n
(D) One——, Two——, Three\n

19. Which of the following programs translates all program instructions at one time and produces a stand-alone object program that can be executed on its own?

(A) An interpreter
(B) A compiler
(C) An assembler
(D) A generator

20. The compilation process can be broken up into THREE main stages. The CORRECT order of the stages is

(A) syntax analysis, lexical analysis, code generation
(B) lexical analysis, code generation, syntax analysis
(C) lexical analysis, syntax analysis, code generation
(D) code generation, lexical analysis, syntax analysis

21. Consider the following partial declaration in C

```
#include<stdio.h>
#define SIZE
main ( )
{
int k;
int table [SIZE];
```

Which of the following lines of code will read FIVE integers from the standard input?

(A)
```
for (k = 0; k < SIZE; k++)
        printf ("%d", table [k]);
```

(B)
```
for (k = 0; k < SIZE; k--)
        printf ("%d", table [k]);
```

(C)
```
for (k = 0; k < SIZE; k--)
        Scanf ("%d", table [k]);
```

(D)
```
for (k = 0; k < SIZE; k++)
        scanf ("%d", table [k]);
```

IF YOU FINISH BEFORE TIME IS CALLED, CHECK YOUR WORK ON THIS TEST.

FORM SPEC 02215010/SPEC/2008

C A R I B B E A N E X A M I N A T I O N S C O U N C I L

ADVANCED PROFICIENCY EXAMINATION

COMPUTER SCIENCE

SPECIMEN PAPER

UNIT 1 - FUNDAMETALS OF COMPUTER SCIENCE

PAPER 02

2 ½ hours

INSTRUCTIONS TO CANDIDATES

1. DO NOT open this examination paper until instructed to do so.
2. Answer **ALL** questions from the **THREE** sections

SECTION A

MODULE 1: COMPUTER ARCHITECTURE AND ORGANISATION

Answer BOTH questions.

1. (a) Differentiate between EACH of the following pairs as they pertain to computers and information systems:

(i) ROM and EPROM **[2 marks]**

(ii) EPROM and EEPROM **[2 marks]**

(iii) MAN and LAN **[2 marks]**

(iv) Mainframe computer and microcomputer **[2 marks]**

(v) Workstation and supercomputer **[2 marks]**

(vi) Cache memory and RAM **[2 marks]**

(b) State how EACH of the following devices can help protect and preserve the resources of a computer:

(i) Surge protector **[4 marks]**

(ii) (Uninterruptible Power Supply) UPS **[2 marks]**

(c) Calculate the following:

(i) The eight-bit signed-magnitude representation of +10 using sign + magnitude. **[2 marks]**

(iii) The decimal of 00001101 (binary). **[1 mark]**

(d) Given the number 8.312×10^5 in standard form:

(i) Identify the exponent **[1 mark]**

(ii) Identify the base or radix **[1 mark]**

(e) Determine if the result of 0011 + 1100 can be stored as a four bit binary number.

[2 marks]

Total 25 marks

2. (a) Briefly explain what is meant by the term 'truth table'. **[2 marks]**

(b) Name the THREE primary logic gates. **[3 marks]**

(c) For TWO of the primary logic gates named in 2(b) sketch the symbol for EACH gate and give the corresponding two input truth table for the gate. **[6 marks]**

(d) Draw the **minimum** logic diagram to show how these primary logic gates can be used to build the Exclusive – OR. **[4 marks]**

(e) (i) Name the TWO **major** components of the Central Processing Unit (CPU). **[2 marks]**

(ii) For EACH of the components named in (e) (i) above briefly describe its function. **[4 marks]**

(iii) Explain what is meant by EACH of the following terms:

(a) Word length **[2 marks]**

(b) Clock speed **[2 marks]**

TOTAL 25 marks

SECTION B

MODULE 2: PROBLEM-SOLVING WITH COMPUTERS

Answer BOTH questions

3. (a) Define the term '*algorithm*'. **[2 marks]**

(b) Structured algorithms and programs are designed using three basic control constructs. Name and briefly describe EACH of these constructs. **[6 marks]**

(c) The problem solving process with computers consists of a sequence of sections that fit together depending on the type of problem to be fixed.

(i) List FOUR stages of the problem solving process. **[4 marks]**

(ii) Select THREE of the stages listed in (c) (i) and describe EACH stage. **[9 marks]**

(d) The following algorithm is supposed to print the squares on the numbers between 1 and 100 (inclusive). However, it is not generating the derived output. Find and correct the problem in the algorithm.

```
Start  = 1
   x   = 1
  while start c = 100
         print start * start
         x   = x + 1
endwhile
```

[4 marks]

TOTAL 25 marks

4. (a) A soft-drink manufacturer sells five soft-drink flavours: bananas, cherry, mango, orange and pineapple. The company knows that banana and pineapple are the two best-selling flavours and is carrying out a poll to determine which of these two flavours is the more popular among its customers

Write an algorithm to find the MORE popular flavour and the NUMBER of votes it obtained. Assume that 100 customers participate in the poll and that there is NO tie. **[15 marks]**

(b) Trace through the execution of the following algorithm and draw the output in your answer booklet exactly as would be generated by the algorithm. You should carefully note the following;

- *printSpaces(n)* prints *n* spaces from the current cursor position
- *print* continues output on the current line form the current cursor position
- *println* causes output on the current line to be terminated, and subsequent output begins on a new line.

```
set SIZE = 10

for i = 1 to SIZE
   print (“*”)
endfor
println ()

for i = 1 to 3
   print (“*”)
   printSpaces (SIZE – 2)
   printlin (“*”)
endfor

for i = 1 to SIZE
   print (“*”)
endfor
println ()
```

[10 marks]

TOTAL 25 marks

SECTION C

MODULE 3: PROGRAMMING

Answer BOTH questions from this section

5 (a) You are employed as a programmer by a software house. You want to learn a new programming language at a nearby community college. You would like the company to pay the cost of the course. Unfortunately, your supervisor sees no reason for the company to pay for the course. His argument is that you already are well versed in one programming language, say PL. He sees no reason for you to learn another. However, he has suggested that you write to the managing director to get approval to do this course. What arguments could you use to convince the managing director that knowledge of a single programming language is not sufficient? **[10 marks]**

(b) Explain the concept of 'structured programming.' **[5 marks]**

(c) State FOUR advantages of using a modular approach in programming. **[4 marks]**

(d) Assume that you have been given a library implementing a list of positive integers. The library supplies you with the following functions:

is_empty
first_element
rest

All functions take a list as input. The first function returns true if the list is empty (contains no elements) and false otherwise; the second function returns the first element in the list (assuming of course that the list is not empty); the third function returns a list consisting of all the elements of the input list except for the first element. Thus, if we pass the list [1 2 3 4] into the function rest, then we get the list [2 3 4] as output.

Write an iterative function to count the number of elements in the list. **[6 marks]**

Total 25 marks

6. (a) Describe FOUR major stages in the compilation process. **[12 marks]**

(b) An integer array, P, contains m positive integer values. The array is sorted in ascending order. An integer array, Q, contains n positive integer values. This array is also sorted in ascending order. Write code in procedural C to merge the integers in P with the integers in Q. The result of the merge is placed into another integer array, R, such that the integers in R are in **ascending** order. Show ALL relevant declarations. Assume that P and Q are already loaded with integers. **[13 marks]**

TOTAL 25 marks

END OF TEST

C A R I B B E A N   E X A M I N A T I O N S   C O U N C I L

ADVANCED PROFICIENCY EXAMINATION

SPECIMEN PAPER
MULTIPLE CHOICE QUESTIONS
FOR

COMPUTER SCIENCE - UNIT 2

Paper 01

READ THE FOLLOWING DIRECTIONS CAREFULLY

Each item in this test has four suggested answers lettered (A), (B), (C), (D). Read each item you are about to answer and decide which choice is best.

.

Sample Item

Which of the following statements is true about objects and classes?

(A) Objects may contain only methods, while classes can have both variables and methods
(B) Classes may only contain methods, while objects can have both variables and methods
(C) A class is an instance of an object
(D) An object is an instance of a class

Sample Answer

(A) (B) (C) ●

The best answer to this item is "An object is an instance of a class", so answer space (D) has been shaded.

There are 21 items in this specimen paper. However, the Paper 01 test consists of 45 items. You will have 120 minutes to answer them.

---

1. An ADT is

(A) the logical properties of data
(B) a combination of data types within a record
(C) a combination of data object and associated operations
(D) a list of variables

2. Which ADT is BEST described as a LIFO structure?

(A) A queue
(B) A variable
(C) A stack
(D) A pointer

Item **3** refers to the following queue of jobs waiting to be printed.

Job1	Job2	Job3	Job4	Job5
Front				Rear

3. What would the queue look like when **three** jobs have been printed and **two new** jobs, Job6 and Job7 have joined the queue?

(A)

Job3	Job4	Job5	Job6	Job7
Front				Rear

(B)

Job4	Job5	Job6	Job7
Front			Rear

(C)

	Job7	Job6	Job1	Job2
Front				Rear

(D)

Job7	Job6	Job1	Job2	Job3
Front				Rear

4. Consider a circular Q with a pointer **front** that references the location of the front of the queue. Assuming that the locations range from 0 to 4, what is the value of **front** after performing the following operations on the queue Q?

enqueue
enqueue
dequeue
enqueue
enqueue
dequeue
enqueue

(A) 0
(B) 1
(C) 2
(D) 3

Item **5** refers to the following pseudocode.

```
x = 1;
i = 1;
while (x < 10)
begin
   x = 2x;
   i = i + 1;
end;
```

5. What is the value of i at the end of the pseudocode?

(A) 4
(B) 5
(C) 6
(D) 7

6. Which of the following combinations is a stack operation?

(A) Enqueue; push;
(B) Dequeue; pop;
(C) Pop; enqueue;
(D) Push; pop;

7. The binary search is BEST described as

(A) a search of only a section of a given list of elements
(B) a search which takes a divide and conquer approach
(C) being very fast when using a very short list
(D) a search of un-order elements

8. Which of the following terms represents types of software maintenance?

I. Adaptive
II. Corrective
III. Defective
IV. Perfective

(A) I, II and III only
(B) I, II and IV only
(C) I, III and IV only
(D) II, III and IV only

9. The MAIN goal of component testing is to

(A) expose faults in the components
(B) demonstrate that the system meets its requirements
(C) increase the supplier's confidence that the system meets requirements
(D) test how one component interacts with others in the system

10. Which of the following is NOT a tool in a software testing workbench?

(A) Test data generator
(B) Report generator
(C) Test comparator
(D) Test manager

11. Which of the following is TRUE of the waterfall approach to the software development process?

I. Cascades from one phase of the process to another.
II. The following stage should not start before the previous has finished.
III. The following stage starts before the previous stage has finished.
IV. The project is partitioned into distinct stages before the phases begin.
V. This approach caters to changing user requirements.

(A) I, II and IV only
(B) I, II and V only
(C) III, I and V only
(D) III, IV and V only

12. The software process model defined by its approach of refining an initial system based on customer input is the

(A) waterfall approach
(B) evolutionary development
(C) formal transformation
(D) reuse-oriented approach

13. What are the TWO phases of the requirements engineering process (software specification)?

(A) Feasibility study and user and systems requirements documentation
(B) Feasbility study and production of interface specification
(C) Interface specification and data structure specification
(D) Sub-system testing and maintenance

14. Which of the following system models depict data transformation as data is processed?

(A) Data dictionaries
(B) Entity-relationship diagrams
(C) Class diagrams
(D) Data flow diagrams

GO ON TO THE NEXT PAGE

15. Which transmission media is BEST suited for transfer of large volumes of data over continents?

(A) Fibre optic cables
(B) Microwave
(C) Coaxial cable
(D) Satellite

16. Which of the following sequences of terms are the MOST related to each other?

(A) Coaxial cables, electronic pulses, strands of glass
(B) Fiber optic cable, light pulses, LAN
(C) Twisted pair cable, conventional telephone
(D) Twisted pair cable, fast and noise resistant, voice and data transmission

17. Which of the following BEST represents the benefits of ALL types of network?

(A) Sharing of peripheral devices, sharing of programs and data, better communication, increased access to database
(B) Sharing of peripheral devices, sharing of programs and data, communication over a wide geographical area, increased access to databases
(C) Sharing of peripheral devices, sharing of programs and data, improved data security, increased access to databases
(D) Sharing of programs and data communication over a wide geographical area, improved data security, increased access to databases

18. What is the name of the device that uses the half-duplex method of transmission?

(A) Television
(B) Radio
(C) Walkie talkie
(D) Telephone

19. Accessing a website via a web browser is an example of which type of network configuration?

(A) Peer-to-peer
(B) Client-server
(C) Peer-to-server
(D) Server-to-server

20. The use of a switched line for data communications offers the user

(A) greater flexibility than does the use of a leased line
(B) the ability to connect to other computers without a modem
(C) less flexibility than does the use of a private line
(D) access to more facilities than the use of dial-up

Item **21** refers to the following statements.

I. Smaller computer systems communicate with one another through the host.
II. A signal is broadcast to all the nodes but only the destination node responds to the signal.
III. A disadvantage of this topology is, if connection is broken, the entire network stops working.

21. To which network topologies does EACH of the above statements refer?

(A) I - STAR II - RING III - BUS
(B) I - RING II - STAR III - BUS
(C) I - RING II - BUS III - STAR
(D) I - STAR II - BUS III - RING

IF YOU FINISH BEFORE TIME IS CALLED, CHECK YOUR WORK ON THIS TEST.

FORM SPEC 02215010/SPEC/2008

C A R I B B E A N   E X A M I N A T I O N S   C O U N C I L

ADVANCED PROFICIENCY EXAMINATION

COMPUTER SCIENCE

SPECIMEN PAPER

UNIT 2 – FURTHER TOPICS IN COMPUTER SCIENCE

PAPER 02

2½ hours

INSTRUCTIONS TO CANDIDATES

1. DO NOT open this examination paper until instructed to do so.
2. Answer **ALL** questions from the **THREE** sections.

SECTION A

DATA STRUCTURES

Answer BOTH questions.

1. (a) Define an ‘Abstract Data Type (ADT)’. **[2 marks]**

(b) (i) Identify ONE primary characteristic of an Abstract Data Types. **[1 mark]**

(ii) Identify ONE specific characteristic of a stack. **[1 mark]**

(iii) Identify ONE specific characteristic of a queue. **[1 mark]**

(c) Write a function called **popStack** in the programming language **C**, which performs the popping operation of a stack. The stack is represented by a one dimensional integer array of 100 elements called **stack**, which must be passed to the function as a parameter. An integer value called **number**, which identifies (the top number of elements) in the stack must also be passed to the function as a parameter. In popping the stack, the function should replace the emptied array location with the value of “**-99-**” and return the value which was popped. **[10 marks]**

(d) Write a function called **dequeue** in the programming language **C**, which performs the dequeue operation of a queue. The queue is represented by a one dimensional integer array of 100 elements called **queue**, which must be passed to the function as a parameter. The function should replace the emptied array location with the value of “**-99**” and return the value which was dequeued. The empty location of the array should be set to “**-99**”. **[10 marks]**

Total 25 marks

2. (a) Explain the principles of EACH of the following techniques.

(i) Bubble sort **[3 marks]**

(ii) Binary search **[3 marks]**

(b) Outline in detail, using narrative, an algorithm for bubble sorting integers into ascending order. **[10 marks]**

(c) A linear or sequential search involves the examination of EACH element in an array. This can be very slow if the desired element is near the end of the list. A better method is to make use of a binary search. This involves storing the data in order as a binary tree. Illustrate, with a diagram, how it is possible to search and find any element in a list of 1000 by examining at most only ten of the elements. **[9 marks]**

Total 25 marks

SECTION B

SOFTWARE ENGINEERING

Answer BOTH questions

3. (a) (i) Name FOUR attributes of a well-engineered software product and describe EACH of them.

(ii) For EACH of the FOUR attributes identified in (a) (i) briefly explain its importance to the software. **[8 marks]**

(b) Identify THREE weaknesses of an unstructured approach to software development. **[3 marks]**

(c) State TWO features of a good software design. **[2 marks]**

(d) Develop a context level diagram for a DVD rental store which engages in the following activities: rents DVDs to its members, receives returned DVDs and purchases new DVDs from suppliers. The sales clerk provides monthly reports to management. **[12 marks]**

Total 25 marks

4. (a) Explain the terms ‘software process’ and ‘software process model’. **[2 marks]**

(b) The Waterfall model and Evolutionary development are two of many generic process models.
Name and briefly describe FOUR fundamental activities which are common to ALL process models. **[8 marks]**

(c) Differentiate between ‘functional requirements’ and ‘non-function and requirements’. **[4 marks]**

(d) Various types of tests are carried out during the validation phase of software development.

Briefly describe EACH of the following types of tests.
(i) Unit
(ii) System
(iii) Acceptance. **[6 marks]**

(e) List FIVE items which should be included in the structure of a test plan. **[5 marks]**

Total 25 marks

SECTION C

OPERATING SYSTEMS AND COMPUTER NETWORKS

Answer BOTH questions

5. (a) Distinguish between 'intranet' and 'extranet'. **[3 marks]**

(b) Describe 'TCP' and 'IP' as they relate to the TCP/IP protocol. **[3 marks]**

(c) (i) Draw a diagram to show the layers of the OSI model for computer communication. **[6 marks]**

(ii) Describe the functions of EACH of the bottom THREE layers of the model drawn in 5 (c) (i) above. **[6 marks]**

(d) Explain using an example how deadlock may arise in an operating system. **[3 marks]**

(e) A program called *exec1* is being executed by an operating system before it can be completed. It is interrupted by a small program *exec2*, which contains four instructions.

(i) What determines which of *exec1* or *exec2* will be run? **[1 mark]**

(ii) Assume exec2 is allowed to run. Explain what is done in the operating system to ensure that *exec1* can be properly executed later. **[3 marks]**

Total 25 marks

6. (a) Explain in detail how an operating system can execute THREE processes at the same time. **[8 marks]**

(b) Describe the characteristics of EACH of the following transmission media.

(i) Coaxial cable **[3 marks]**

(ii) Twisted pair **[3 marks]**

(c) Describe TWO network related problems that can be solved by using network diagnostic tools. **[4 marks]**

(d) Explain TWO reasons why compression utilities are useful. **[4 marks]**

(e) Describe 'radio waves' as a transmission media. **[3 marks]**

Total 25 marks

END OF TEST

C A R I B B E A N   E X A M I N A T I O N S   C O U N C I L

CARIBBEAN ADVANCED PROFICIENCY EXAMINATION

COMPUTER SCIENCE - Unit 1

SPECIMEN PAPER 2008

Item Number	Key
MODULE 1	
1	A
2	C
3	D
4	C
5	A
6	B
7	A
MODULE 2	
8	B
9	D
10	C
11	C
12	A
13	A
14	D
MODULE 3	
15	B
16	C
17	D
18	A
19	B
20	C
21	D

FORM SPEC 02115010/SPEC/2008

C A R I B B E A N   E X A M I N A T I O N S   C O U N C I L

ADVANCED PROFICIENCY EXAMINATION

COMPUTER SCIENCE

SPECIMEN PAPER

UNIT 1

PAPER 02

MARK SCHEME

Question 1

(a) (i) Instructions in ROM (1) cannot be changed, but the contents of EPROM can be erased using special equipment (1). **[2 marks]**

(ii) EPROM chips must be removed from the computer to be changed but EEPROM chips can be changed by using special electrical impulses within the computer (1).

EPROM normally has to be removed from a computer before it can be reprogrammed while EEPROM can be reprogrammed without being removed from the computer (1). **[2 marks]**

(iii) MAN is a network that spans a city area (1) and LAN is a network that spans a small area like a building. (1) **[2 marks]**

(iv) Main frame computers are bigger in size (1), more expensive, have much more memory than microcomputers. (1) **[2 marks]**

(v) A supercomputer is much more powerful than a workstation computer. (1) Which can handle less complex tasks (1). **[2 marks]**

(vi) Cache memory is much faster than RAM (1) and consequently more expensive (1) while RAM is similar to cache memory. **[2 marks]**

(b) (i) Surge protectors protects the computer (1) equipment from surges in power (1). The protector allows multiple components (1) to be plugged into one power outlet. (1) **[4 marks]**

(ii) UPS – propose is to convert power and store it in a battery OR runs on battery if the power goes off, to give users time to shut down the computer equipment. **[2 marks]**

(c) (i) +10 = 000010140
eight digits **[1 mark]**
correct answer 00001010 **[1 mark]**

(ii) The decimal equivalent of 00001101 is + 13 **[1 mark]**

(d) (i) exponent: 5 **[1 mark]**

(ii) base or radix: 10 **[1 mark]**

(e) 0011
1100
1111 Result **[1 mark]**

Yes it can be stored as a from digit binary number **[1 mark]**

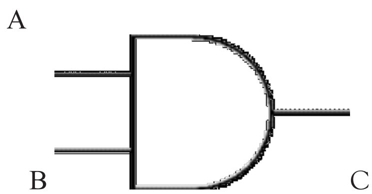
Module 1 – Specific Objectives: 1, 3

Question 2

(a) A truth shows the output (1) to various combinations of logic input (1) **[2 marks]**

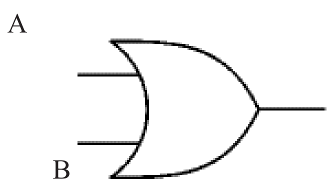
(b) Three primary topic gates are
NOT
AND
OR
[3 marks]

(c)



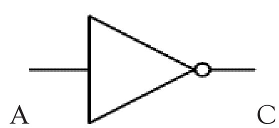
AND Gate

A	B	C
1	1	1
1	0	0
0	1	0
0	0	0



OR Gate

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1



NOT Gate

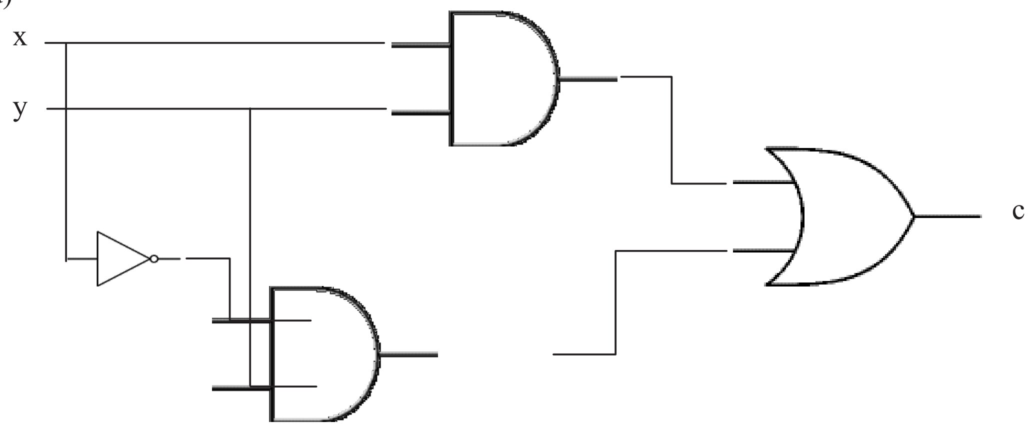
A	C
0	1
1	0

1 mark for each symbol
2 marks for correct truth table
1 mark for partial completion of truth table

[6 marks]

Question 2 cont'd

(d)



[4 marks]

(e) (i) Control Unit (CU) (1) and Arithmetic and Logic Unit (ALU). (1)

(ii) The CU controls the movement of data (1) and instructions (1) into and out of the CPU and controls the operation of the ALU (1).

The ALU does the actual computation or processing of data (1). It performs all arithmetic and logic operations (1).

[2 marks]

(iii) (a) Word length – the <u>number of bits</u> the <u>CUPCAN process</u> at any one time

[2 marks]

(b) Check speed – an <u>internal of time</u> in <u>the processor</u>

Module 1 – Specific Objective: 2

Question 3

(a) An algorithm is a sequence of unambiguous steps (1) for solving a problem (1). **[2 marks]**

(b) Sequential (1): Steps are performed in a strictly sequential manner, each step being executed exactly once (1).

Selection (1): One of several alternative actions is selected and executed. (1)

Repetition (1): One or more steps is performed repeatedly (1). **[6 marks]**

(c) (i) (1) Problem Definition.
(2) Problem Analysis.
(3) Generating possible Solutions.
(4) Analyzing the Solutions.
(5) Selecting and Justify Solutions.

Any four 4 marks

The process is only a guide for problem solving. It is useful to have a structure to follow to make sure that nothing is overlooked. Nothing here is likely to be brand new to anyone, but it is the pure acknowledgement and reminding of the process that can help the problems to be solved.

(ii) 1. Problem Definition

The process for solving a problem will initially involve defining the problem (1) you want to solve. You need to decide what you want to achieve and write it down. The first part of the process not only involves writing down the problem to solve, but also checking that you are answering the right problem (1). It is a check-step to ensure that you do not answer a side issue or only solve the part of the problem that is most easy to solve. People use the most immediate solution to the first problem definition that they find without spending time checking the problem is the right one to answer (1).

2. Problem Analysis

This step involves determining what are the inputs (1), what processing is involved (1) and what are the outputs (1). The next step in this process is used to check where we are and what is the current situation. Also determine how the data and information or results will be stored temporarily or for future use.

3. Generating possible Solutions

In this stage you concentrate on generating many solutions (1) and not evaluate them at all (1). Very often an idea, which would have been discarded immediately, when evaluated properly, can be developed into a superb solution. At this stage, you should not pre-judge any potential solutions (1) but should treat each idea as a new idea in its own right and worthy of consideration.

Question 3 cont'd

(c) (ii) 4. Analyzing the Solutions

This section of the problem solving process is where you investigate the various factors about each of the potential solutions (1). You note down the good and bad points and other things which are relevant to each solution (1). Even at this stage you are not evaluating the solution because if you do so then you could decide not to write down the valid good points about it because overall you think it will not work. However you might discover that by writing down its advantages that it has a totally unique advantage (1).

5. Selecting and Justify Solutions

This is the section where you look through the various influencing factors (1) for each possible and decide which solutions to keep (1) and which to disregard (1). You look at the solution as a whole and use your judgement as to whether to use the solution or not. Sometimes pure facts and figures dictate which ideas will work and which will not.

(d) (i) Infinite loop (1): The number 1 is printed repeatedly (1)

(ii) Replace the statement y = y+1 with x = x+1 (1)

Total 25 marks

Module 2 – Specific Objectives: 2, 3, 6, 8

Question 4

(a)

```
{ bCount   =   pCount     0                                   [1 mark]

  for j   =  1 to 100 do                                      [1 mark]
    input choice
    if choice   =  'banana' then                              [1 mark]
      bCount  =  bCount + 1                                   [1 mark]
      else
    pCount   =  pCount + 1 {Assumes all choices are valid}    [1 mark]
    endif                                                     [1 mark]
  endfor

  if bCount > pCount then
     print  "The most popular flavour is banana."             [1 mark]
     print  "Votes:", bCount                                  [1 mark]
  else
     print  "The most popular flavour is pineapple."          [1 mark]
     print  "Votes:", pCount                                  [1 mark]
  endif
```

Total 10 marks

(b) Output generated by the algorithm is as follows:

```
* * * * * * * * * *
*                 *
*                 *
*                 *
* * * * * * * * * *
```

If the above diagram is drawn exactly as shown:

If there are variations, use the following mark scheme:

All of top line: **[2 marks]**
Some of top line: **[1 mark]**

All of left side of rectangle: **[2 marks]**
Some of left side of rectangle: **[1 mark]**

All of right side of rectangle: **[2 marks]**
Some of right side of rectangle: **[1 mark]**

All of bottom line: **[2 marks]**
Some of bottom line: **[1 mark]**

Top line properly connected to lower part: **[2 marks]**
Bottom line properly connected to upper part: **[1 mark]**

Total 10 marks

Module 3 – Specific Objectives: 7, 9

Question 5

(a) The development of a program essentially involves two tasks

(i) Program design – designing a solution (1)

(ii) Coding – writing code in a programming language (1)

There are several strategies used to develop a design, each strategy required that the programming language used to code the program support particular features (1). Therefore knowing just one language locks the programmer into using particular strategies (1). This is undesirable since the strategy selected to design the program should be based on the type of programming problem rather than the eventual language of implementation (1).

Also, once the program is designed a programmer should be able to choose among a variety of languages since even within a particular class of languages, individual language differs. The programmer should be able to choose the most appropriate language, that is the language that provides:

(i) Ease of programming particular solution (available predefined data structures and subroutines) (1)

(ii) Run-time efficiency – some languages are inherently more efficient than others (1)

(iii) Portability (1)

(iv) Ease of debugging (1)

(v) Maintainability (1)

[10 marks]

(b) Structured programming is an approach to programming with can give higher productivity (1) and which can produce programs of high quality which are easy to test, debug, modify and maintain. (1) The approach emphasizes modular development of a program, that is, a large problem is divided into smaller subdivided and so on. (1) For each subproblem, a module is developed to solve it. (1) The approach recommends the use of a few simple control structures in specifying algorithms, also GOTO's should not be used so that program would be easier to understand and debug. (1) **[5 marks]**

(c) (i) Individual modules can be tested separately (1)

(ii) Modules can be kept in a library and be reused in other programs (1)

(iii) Programs are easier to build, read, debug and maintain (1)

(iii) The modular approach means that several programmers can each work on separate module, thus shortening the total development time of a large project (1). **[4 marks]**

Question 5 cont'd

(d)

```
Int number_of_elements (List I)          [1 mark]
{
int ret = 0                              [1 mark]
while (!is_empty (I))                    [1 mark]
{
        return ++;                       [1 mark]
                      I = rest(I);       [1 mark]
}
        return ret;                      [1 mark]
}
```

[6 marks]

Module 3 – Specific Objectives: 1, 9

Question 6

(a) Lexical analysis (1) - Lexical analysis scans (1) the characters of the same program from left to right (1) and builds the actual symbols of the program – integers, identifiers, reserved words, etc. (1)

Semantic analysis (1)- Takes constructs and checks them for semantic correctness (1), and stores the necessary information (1) about the constructs in the symbol table. (1)

Code optimization (1)- Reduces execution time (1) for project by looking at how the code (1) could execute more efficiently. (1)

Code generation (1) - Actual translation of the internal source program (1) into assembly language (1) or machine language. It is the most detailed part of the compilation. (1)

1 mark for EACH stage
2 marks for EACH explanation
[12 marks]

Question 6 cont'd

(b)

```
#include <stdio.h>

void main ( )
{  int a  [ ]  =  {2, 3, 4, 4, 4, 6, 7, 8, 10, 12};        [1 mark]
int b  [ ]  =  {1, 3, 5, 10, 11, 13, 14, 15};               [1 mark]
int c  [25];                                                [1 mark]
int i  =  0;                                                [1 mark]
int m  =  10,  n  =  8;
int j  =  0,  k  =  -1;
while  (i  <  m  &&  j  < n)
        {  k++;
           if  (a[i]  <  b  [j])                            [1 mark]
        {  c  [k]  =  a  [i];                               [1 mark]
          i++;                                              [1 mark]
        }                                                   [1 mark]
        else { c  [k]  =  b  [j];
                j ++;
              }
         }
        int t;

        if  (  i  = =  m  )  //end of array a
          for  (t  =  j;  t  <  n;  t++)                    [1 mark]
             {  k++;                                        [1 mark]
                c  [k]  =  b  [t];

             }
        else
        if  (  j  = =  n  )  // end of array b              [1 mark]
          for  (t  =  i;  t  <  m;  t++)                    [1 mark]
             {  k++;
                c  [k]  =  a  [t];                          [1 mark]
             }
```

TOTAL 13 marks

Module 3 – Specific Objectives: 3, 8

C A R I B B E A N   E X A M I N A T I O N S   C O U N C I L

CARIBBEAN ADVANCED PROFICIENCY EXAMINATION

COMPUTER SCIENCE - Unit 2

SPECIMEN PAPER 2008

Item Number	**Key**
MODULE 1	
1	C
2	C
3	A
4	C
5	B
6	D
7	B
MODULE 2	
8	B
9	A
10	C
11	A
12	B
13	A
14	D
MODULE 3	
15	D
16	C
17	A
18	C
19	B
20	A
21	D

FORM TP 02215010/SPEC/2008

C A R I B B E A N E X A M I N A T I O N S C O U N C I L

ADVANCED PROFICIENCY EXAMINATION

COMPUTER SCIENCE

SPECIMEN PAPER

UNIT 2

PAPER 02

MARK SCHEME

Question 1

(a) The combination of a data structure and a group of functions or procedures (1) designed to manipulate the data within the data structure (1). **[2 mark]**

(b) (i) An ADT must possess the following characteristics:

1. the facility to create a container to hold the data;
2. the facility to add a new element to the container;
3. the facility to remove/delete an element which satisfies some criterion form the container;
4. the facility to find an element which satisfies some criterion within the container;
5. the facility to destroy the container when it is no longer required.

[Any one 1 mark]

(ii) a stack
The specific operation which defines a stack is the concept of "Last-in-first-out". (1) Data elements which have been added to the container are removed from the container in the reversed order (1) to which they were inserted. (1) Typically, this ADT is identified by a variable called "the head" (1) which identifies the location of the most recent element (1) added to the container. The functions of "push" and "pop" are typically used to describe the process of inserting and deleting elements respectively, from the stack.
[Any one 1 mark]

(ii) a queue
The specific operation which defines a queue is a concept of "Fist-in-first-out". (1) Data elements which have been added (1) to the container are removed from the container in exactly the same order (1) in which they were inserted. Typically, this ADT is identified by a variable called "the head" (1) which identifies the location of the longest existing element in the container, (1) and another variable called "tail" which identifies the most recent element added to the container. (1) The functions of "enqueue" and "dequeue" are typically used to describe the process of inserting and deleting elements respectively, (1) from the queue. In the queue, data elements are inserted through the tail and removed from the head. (1)
[Any one 1 mark]

Question 1 continued

(d)

```
int popStack(int stack[], int number)
        {
                int value;

/* Check for error situation first */
                if (number < 1)
                        {

printf("The stack is already emty");
                                return -1;
                        }
                else if (number > 99)
                        {

                printf("Elements size is beyond the stack size");
                return -1;


        }
else    /*Stack can be popped */
        {
                value = stack[number -1];
                stack[number-1] = -99;
                number = number -1;
                return value;

        }

}
```

1 mark for data type of return value
1 mark for passing array parameters
1 mark for int parameter number

1 mark for testing stacks empty

1 mark for return <u>message</u>

1 mark for testing
1 mark for message

1 mark for assignment

1 mark for replacing value with -99

1 mark for returning value

Question 1 continued

(e)

```
int dequeue(int queue[])                      1 mark for data type of value to be returned
    {                                         1 mark for passing array as parameter
        int value;

        /* Check for error situation first */       1 mark for test condition
        if (queue [0] = = -99)
            {

                printf("The queue is already empty");
                return -1;
            }
        else    /* Queue can be dequeued */
            {

            value = queue [0];

            /*Shift all the elements one place up */
            for (int x = 0; queue [x] !=-99 && x < 100;x++)   3 marks for conditional
                {                                              loop

                    queue [x] = queue [ x + 1];                1 mark for assignment
                }
            return value;                                      1 mark for return value
        }
```

Module 1 – Specific Objectives: 2, 3, 4

Question 2

(a) (i) A sorting procedure that works by repeatedly stepping through the list of items to be sorted **(1)**, comparing two items at a time and swapping them if they are in the wrong order **(1).** The pass through the list is repeated until no swaps are needed, which means the list is sorted **(1)**. **(3 marks)**

(ii) A search technique for finding a particular value in a sorted list **(1)**. The desired item is compared to the data in the middle of the list **(1)**. The half that contains the data is then compared in the middle, and so on, either until the key is located or a small enough group is isolated to be sequentially searched **(1)**. **(3 marks)**

(b) An example using N(i), where i is the position of the ith element in the list.

(i) **Start** with the first pair of numbers in the list **(1)** and **compare** them **(1)**, that is, compare N(i) with N(i+1).

(ii) If N(i) greater than N(i+1), then **swap (1)**. That is, N(i) becomes N(i+1) and N(i+1) becomes N(i). If a swap occurs, make note of it,that is, set **flag** to indicate swap occurred **(1)**.

(iii)Take next pair of numbers, that is, **increment i (1)** and **repeat steps (i) and (ii)**, until the last pair if number in list have been compared **(1)**.

(iv)If a swap took place, that is, the flag was set, then **clear flag (1)** and **repeat the entire procedure (1)**.

(v) If **no swap** occurred during any iteration **(1)**, then **end (1)**. **(10 marks)**

(c) Binary tree (Note: only One part of subtree shown):

Search:	1000	
1st	500	500
2nd	250	250
3rd	125	125
4th	63	62
5th	31	32
6th	15	16
7th	7	8
8th	3	4
9th	1	2
10th	1	1

[1 mark for each correct search, no more than 9 marks]
(3 x 2 marks = 6 marks)

Module 1 – Specific Objectives: 5, 6

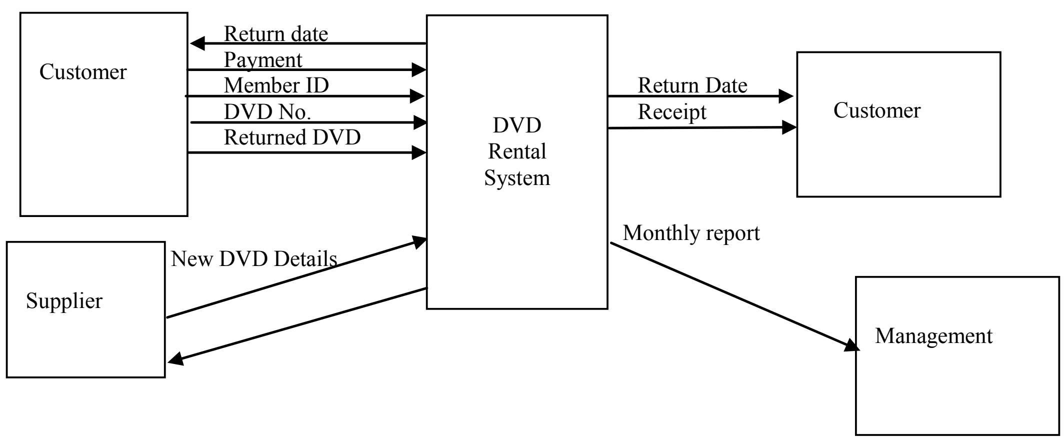
Question 3

(a) Maintainability: Software should evolve to meet the changing needs of customers.
Dependability: Software should be reliable, secure and safe.
Efficiency: Software should not make wasteful use of system resources.
Usability: Software should have appropriate user interface and adequate documentation. **[2 x 4 = 8]**

(b) Software is likely to be of poor quality and unsuitable.
Software development cost is likely to be higher that expected.
Software development time is increased. **[3 x 1 = 3]**

(c) Correctly implements a specification
Allows efficient code to be developed
Can be adapted to modify functions and to add new functionality. **[Any 2 x 1= 2]**

(d)



1 for each labeled entity
1 for each labeled data flow

Module 2 – Specific Objectives: 2, 4

Question 4

(a) A software process is a structured set of activities required to develop a software system. (1)

A software process model is an abstract representation of a software process. (1) **[2 marks]**

(b) *Software specification:*
The process of establishing what services are required and the constraints on the system's operation and development.

Software design:
The process of converting the system specification into an executable system.

Software validation:
Software is checked to ensure that it does what the customer wants.

Software evolution:
Software is modified to adapt to changing customer needs.

1 mark for Name
1 mark for explanation [2 x 4] **[8 marks]**

(c) *Functional requirements*
Statements of services the system should provide, (1) how the system should react to particular inputs and how the system should behave in particular situations. (1)

Non-functional requirements
The constraints on the services or functions (1) offered by the system such as timing constraints, constraints on the development process and standards. (1) **[4 marks]**

(d)

(i)	Unit testing:	Individual components (1) are tested independently. (1)
(ii)	System testing:	Testing of the system as a whole. (1) Testing of emergent properties is particularly important. (1)
(iii)	Acceptance testing:	Testing with customer data to check that the system meets the customer's needs. (1)

[6 marks]

(e)
- The testing process
- Requirements traceability
- Tested items
- Testing schedule
- Test Record procedures
- Hardware and Software requirements
- Constrains

[5 marks]
[Any 5 x 1 mark each]

Module 2 – Specific Objectives: 3, 7

Question 5

(a) (i) An intranet allows persons within an organisation to access information (1). Where as an extranet is an intranet that has been extended (1) to allow selected external organizations to access information (1). **[3 marks]**

(b) TCP is a reliable transport protocol (1) with retransmissions etc. (1). Where as the IP is a network layer protocol which can be unreliable (1). **[3 marks]**

(c) (i)

Application
Presentation
Session
Transport
Network
Data Link
Physical

One mark for each layer, correctly positioned **[6 marks]**

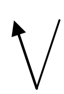
(ii) Network: routing packets (1) congestion control (1).

Data Link: error control (1). Communication between 2 nodes (1).

Physical: bits->cable signals(1). How much volts is a 0/1 (1). **[6 marks]**

(d) Process A needs a resource that process B is using (1) while process B needs a resource that process A is using (1). Neither process can proceed (1) until the desired resource is obtained, resulting in a deadlock (1).

A->B



[3 marks]

(e) (i) Priorities of each program (1). [1 mark]

(ii) The state of *exec1* is saved (1) **before** switching to *exec2* (1). After *exec2* completes (1) *exec1's* state is restored and execution continues(1). **[3 marks]**

Total 25 marks

Module 3 – Specific Objectives: 3, 4

Question 6

(a) (i) Time-slicing in a round-robin fashion is normally used (1). If process A is running initially, it is pre-empted (1) and its state changed and saved (1). Then process B can be loaded (1) and run and so on (1). When process A's turn comes again, its state and information is restored (1) and it is executed for a short time again (1). This continues until all processes are completed (1). **[8 marks]**

(b) (i) Coaxial cable – has a bandwidth that exceeds 100 Mbps (1). Its typical use is for transmitting cable TV signals (1). Coaxial cable has a copper wire core (1). **[3 marks]**

(ii) Twisted pair – this cable usually contains 4 wires (1). Each wire is coated in plastic, so the copper wires do not come in direct contact with each other (1). It is available as unshielded twisted pair and shielded twisted pair (1). **[3 marks]**

(iii) Fiber-optic cables – consist of a bundle of extremely thin tubes of glass (1). Each tube is much thinner than a human hair. Data is sent using optical signals (1). These cables are not prone to electromagnetic interference (1). **[3 marks]**

(c) (1) There may be a cut or kink (1) in the cable, and the diagnostic tools can help locate the location of the fault (1). **[2 marks]**

(2) Diagnostic tools can help detect malfunctioning hardware on the network (1), e.g. network interface cards. (1) **[2 marks]**

(d) Compressed files occupy less space (1), so more files can actually be stored on your hard disk (1). **[2 marks]**

Because compressed files occupy less space (1),. They can be transmitted more quickly across a network than uncompressed files (1). **[2 marks]**

(e) Radio waves are omni-directional, i.e. they spread out in multiple directions when they are transmitted (1). These signals can cover very long distances (1). For radio broadcast, you need a transmitter to send the signals and a receiver to accept the broadcast signal (1). **[3 marks]**

Total 25 marks

Module 3 – Specific Objectives: 3, 4

CARIBBEAN EXAMINATIONS COUNCIL

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION

MAY/JUNE 2004

COMPUTER SCIENCE

COMPUTER SCIENCE

CARIBBEAN ADVENCED PROFICIENCY EXAMINATIONS
MAY/JUNE 2004

INTRODUCTION

This is the fourth year of open examinations for the Unit-one course comprising three papers and the third year for the Unit-two course. Papers 01 and 02 for both units were examined externally by CXC while Paper 03 was examined internally by the teacher and moderated by CXC.

Paper 01 consisted of three sections, each corresponding to a module of the syllabus. There were five compulsory short-response questions within each section, for which a maximum of 50 marks could be obtained, for a total of 150 marks.

There were six questions of equal weighting in Paper 02, two questions per module and candidates were expected to answer one question from each module. These extended-response questions required more depth of understanding than the questions in Paper 01.

The individual contributions of Papers 01, 02 and 03 to the examination remained 50 per cent, 30 per cent and 20 per cent respectively.

GENERAL COMMENTS

Performance in Unit 1 suggests that teachers and candidates are completing the syllabus, and as such candidates are for the most part performing well.

Examiners continue to be very concerned about candidates' performance in Unit 2; in particular Modules 2 and 3 where programming knowledge is tested. Candidates continue to find it extremely difficult to write even simple programs, especially in the object-oriented languages.

However, it appears that teachers are not completing the syllabus for Unit 2, and therefore, candidates are coming to the examination having not acquired all the necessary skills. It seems that candidates are not getting the necessary practice during preparation for the examination. Perhaps this is so because, in many cases, they are coping with learning a new programming language, whilst simultaneously developing a system in order to satisfy the requirements for the internal assessment. For Unit 2, teachers are reminded to have several practical sessions with students before the final exam.

Candidates should be encouraged, as part of their examination technique, to read questions carefully and to respond with sufficient detail to commensurate with the marks indicated.

DETAILED COMMENTS

UNIT 1

PAPER 01

SECTION A – Components of Computer Systems

Question 1

This question tested candidates' knowledge of terms relating to computer hardware. In general, part (a) was poorly answered. Many candidates described clock speed as time taken to perform a task. Few candidates were able to mention the function of cache memory. The main strong point in the responses was the ability of most candidates to correctly identify two problems that the UPS can prevent. In a few cases candidates listed features of the UPS as problems that it can prevent.

Question 2

This question examined candidates' understanding of the resource management functions of an operating system. Candidate response was generally weak. There were no strong points in the responses. Of particular concern was the candidate's lack of knowledge of deadlock, interrupt and channel bandwidth. A bit more depth of coverage of this area of the syllabus is recommended so that candidates can express themselves more clearly.

Question 3

This question tested candidates' knowledge of the reasons why operating system (OS) software needs to be kept up-to-date and the services offered by disk management software.

The average mark for this question was 50 per cent. Most candidates failed to give correct responses to Part (a). They seemed not to recognize that updates are really additions or changes to the original software. Updates are needed to fix errors due to undetected bugs; add new features to the OS; provide new code that can perform tasks more efficiently. There were generally good responses to Part (b). Generally, candidates were able to correctly identify the services offered by disk management software.

Question 4

This question tested candidates' understanding of the functions of components in a network. Candidate responses were generally good. The diagrams and explanations relative to a hub in a network were quite good. The meaning of 'data encryption' was well explained, including the method and reasons. Candidates understood the need for confidentiality and security. Items that are normally encrypted were correctly identified.

Question 5

This question examined candidates' knowledge and understanding of transmission media and network topologies. The overall performance on this question was fairly good. Parts (a) and (b) (i) were well answered. Candidates seemed not to understand what was required for Part (b) (ii). The main weak point was that many candidates' descriptions of the FDDI topology were not reflected in the diagrams that they drew.

SECTION B – Application of Computers

Question 6

This question was designed to test candidates' understanding of the role of computers within an organization. The majority of candidates seemed to have misinterpreted the question. They treated the application areas of the business as Word Processing, Spreadsheets and Database Management. Consequently, responses to Part (b) were generally meaningless.

Question 7

This question tested candidates' knowledge of the various types of information systems. This question was generally well done. Part (a) was well known by most candidates. However a few candidates did not mention at which level in the organization the information system was used. The main weaknesses were in Part (b). Few candidates were able to recognize that an 'executive information system' is similar to a 'decision support system'.

Question 8

This question tested candidates' knowledge of issues relating to telecommuting, piracy and data security. Part (a) was not answered well by a lot of candidates. Most candidates mistook telecommuting for telemarketing and wrote about buying and selling. Part (b) was well done as most candidates knew the implications of pirating software and also the use of firewalls and encryption.

Question 9

This question examined the different types of data security and the strategies for ensuring data security. This seemed to be a popular topic with candidates. The question was very well done. The security controls were well identified and explained.

Question 10

This question tested candidates' knowledge of the benefits and drawbacks of using a computer-based system in an organization. This question was generally well-answered. Most candidates were able to give good responses. Candidates should be encouraged to use examples, descriptions and explanations that are required at this level.

SECTION C – Computer-based Problem Solving

Question 11

This question assessed candidates' knowledge of the use of a computer-based tool in the solution to a real-life problem. The question also tested knowledge of information sources. It was answered very well. The only weak point was that a few candidates did not know the features of the tools that were useful in performing the given task.

Question 12

This question tested candidates' knowledge of the characteristics of information sources. The responses were for the most part reasonable. Candidates were able to correctly identify and discuss characteristics such as *currency*, *relevance*, *cultural context*, *bias*, *accuracy* etc. In part (b), even though candidates knew differences between 'book and newspaper', 'CD-ROM and DVD-ROM', those differences did not relate to information. Surprisingly many could not state similarities such as: *book* *and newspaper* – both use a printed medium *CD-ROM and DVD-ROM* – both based on optical technology; both can store at least 650 MB.

Question 13

This question tested candidates' knowledge and understanding of issues related to online/internet publishing. Part (a) was poorly done. Candidates failed to recognize that the publisher could derive revenue by letting customers pay a fee to subscribe and then each customer gets a username and password. Parts (b) and (c) were fairly well done. Candidates were able to correctly describe advantages and disadvantages to the customer of having the book available online.

Question 14

This question tested candidates' knowledge and understanding of the selection of computer-based tools that are best suited to a given situation. The question was generally well done. In both Parts (a) and (b), the candidates demonstrated that they were familiar with the tools and that they knew the appropriateness of the chosen tool for the task to be performed. In Part (c) there was some doubt as to the precise functions of the built-in grammar checker. For example some did not know that a grammar checker also provided the meaning of words.

Question 15

This question tested candidates' knowledge and understanding of issues relating to different types of information sources. This question was very well done. Candidates were easily able to identify information sources and suggest why they were suitable choices. The difficulty was with Part (c) where the discussion on the disadvantages was limited and most candidates received 2 of the allotted 4 marks for that section.

PAPER 02

SECTION A – Components of Computer Systems

Question 1

This question examined candidates' knowledge and understanding of different types of networks, network configurations and protocols. Generally Part (a) was answered fairly well. Some good diagrams were drawn. However, in some cases candidates' description of the function of the Virtual Private Network (VPN) was limited. They omitted the fact that the VPN devices at the branches translated between the packet format of the Internet and the packet format of the networks at the branches. Part (b) was well known by most of the candidates. Part (c) was poorly done. Many candidates could not show how the peer-to-peer and client/server configurations could co-exist on the same network. Most candidates got the OSI model correct. The weaker candidates were able to draw at most 4 layers in the correct order. The main weak points in the responses were the major cost associated with VPN and the client/server – peer-to-peer diagram.

Responses to this type of question can improve with a better understanding of network types, especially VPN and VAN.

Question 2

This question was designed to test candidates' knowledge of network components and their understanding of the factors that have to be considered when installing networks. This question was poorly done. Most of the candidates got less than 30 percent of the marks allocated. For Part (a) very few could identify the factors to be considered. Even fewer understood what was involved in providing simultaneous Internet access to a number of computers in a network. For Part (b) again only a few candidates correctly identified component A as a *gateway* and component B as a *hub*. The majority of candidates obtained the marks for Part (c).

SECTION B – Application of Computers

Question 3

This question tested candidates' knowledge and understanding of data security issues/problems, and their understanding of expert systems. Responses were generally good. Part (a) was well answered. Candidates demonstrated a good understanding of the data security problems - and the strategies to deal with those problems - which a company may face when sensitive data is held on a computer. In response to Part (b), candidates' definitions of an expert system were not clear. For Part (c), some gave advantages of expert systems such as: hospital can save money by hiring fewer doctors. Those who attempted Part (d), mentioned only one disadvantage, namely that the expert system could misdiagnose and possibly lead to the death of a patient. Another disadvantage is that the system is difficult to keep up-to-date with all recent medical findings.

Question 4

This question tested candidates' knowledge of the use of robots in industry, and issues relating to the use of Automatic Teller Machines (ATM) in banks. The question was for the most part well done. Most candidates were familiar with the use of the robotics, particularly in a vehicle assembly plant. Part (c) was also well done. Candidates demonstrated a good understanding of the ATM. The question intended to ask for one technique that the *bank* could use to protect a customer's account from disclosure while using the ATM. However, many candidates interpreted the question to be requiring one technique that the *customer* can use. Part (a) was the weak point. Candidates did not understand what an expert system was and therefore could not properly answer this part. On the whole this was a very popular question.

SECTION C – Computer-Based Problem Solving

Question 5

This question tested candidates' knowledge and understanding of the stages in the problem-solving process, and their ability to describe types and characteristics of information sources. Very few candidates answered this question to the satisfaction of the examiners. Those who attempted Part (a), succeeded only in listing the stages in the System Development Life Cycle, but were unsure of the deliverables of each stage. Part (b) was fairly well done. The responses to Part (c) showed that candidates had ideas of what was requited but had difficulty expressing their ideas clearly.

Question 6

This question examined candidates' knowledge of the features of modern productivity tools, and their suitability to certain tasks. Overall performance in this question was fair. Part (a) was well done with most of the candidates obtaining more than 75 percent of the allotted marks. Many candidates did seem to understand the question. Many wrote of preventing paper wastage by reducing font size and margins. In Parts (c) and (d), candidates stated the features without full descriptions as to their suitability for the given task.

UNIT 2

PAPER 01

SECTION A – Software and System Development

Question 1

This question tested candidates' knowledge and understanding of the need for a structured approach to software development and the activities that are common to all software development processes. Both sections of the question proved difficult for the candidates. Part (a) was poorly done. Many candidates wrote the stages of the System Development Life Cycle. Even though most candidates performed better in Part (b), the responses given were for the most part verbatim from the recommended texts.

Question 2

This question examined candidates' knowledge of the attributes of well-engineered software, and the effect that lack of such attributes will have on the software. This question was fairly well done by most candidates. Some candidates, however, correctly identified the attributes of well-engineered software, but gave incorrect descriptions. Responses to Part (b) were in many cases generalized and were not related to the attributes stated in Part (a). A comparison chart or attribute table may be instructive in assisting candidates with this area of the syllabus.

Question 3

This question assessed candidates' knowledge of functional and non-functional requirements and the techniques used in the analysis phase to obtain these requirements. This question was very well done.

Question 4

This question tested candidates' knowledge and understanding of the use of Data Flow Diagrams (DFD) during analysis. Part (a) was well done. In Part (b) many candidates copied the symbols as given in the question, but were unable to identify the data flows to complete the DFD. More exposure to DFDs should be provided.

Question 5

This question examined candidates' knowledge and understanding of the activities, tools, techniques and deliverables of the design phase of software development. Most candidates were able to list the activities of the design phase, but had difficulty in providing meaningful explanations. For example many candidates did not include key terms in their responses such as a HIPO chart emphasises the 'relationships' among the elements of a system. Most mentioned only that the HIPO chart showed input, processing and output. Most of the examples used did not reflect a system, but rather a component of a system. Candidates should have more exposure to practical examples of the use of HIPO charts and teachers should ensure that proper annotations are included.

SECTION B – Programming Languages

Question 6

This question assessed candidates' understanding of the term 'algorithm' and their ability to write a simple algorithm. Most candidates were able to explain what was meant by the term 'algorithm'. However, many had difficulty writing the required 'algorithm' for Part (b). Candidates need to be given more practice in the writing of 'algorithms'.

Question 7

This question tested the candidates' knowledge of the characteristics of programming languages, and the ways in which programming languages differ. The question was generally well done, however, many candidates attempted to give too much detail when the question asked for 'brief' descriptions. In their description of the object-oriented programming paradigm, many candidates omitted the fact that the programs consist of a set of communicating objects.

Question 8

This question examined the candidates' knowledge and understanding of programming constructs and tested their ability to describe the function of the constructs using examples in each case. Most of the candidates were able to describe the function of the construct and provide appropriate examples. However, surprisingly, they could not identify the constructs by their names, for example, 'selection', 'iteration', 'sequencing'.

Question 9

This question tested the candidates' understanding of the factors that have to be considered when choosing a programming language for a given application. The question was generally well done. The candidates' main strong points were in identifying the factors to be considered. However, their explanations were far from adequate. Part (b) was poorly done - only a few candidates demonstrated any understanding of the functional programming paradigm.

Question 10

This question tested candidates' understanding of concepts associated with the object-oriented and procedural paradigms. Many candidates had a general idea of the conceptual building blocks of object-oriented design, but only a few demonstrated real mastery. Almost all candidates were able to explain the term 'recursion' with respect to the procedural paradigm. Part (c) was generally well done.

SECTION C – Program Development

Question 11

This question tested candidates' ability to draw the layout of a simple Help system using a Graphical User Interface (GUI), and to identify events that the help system should anticipate when the user interacts with the system. In Part (a) most candidates had difficulty identifying the events and it was clear they were not sure what an event was. For Part (b) it was widely known by candidates that a diagram had to be drawn. However, many candidates did not properly identify the GUI objects in the layout. In some cases candidates thought that designing a layout meant writing a description or writing code.

Question 12

This question tested candidates' knowledge and understanding of abstract data types (ADT) and operations on ADTs. Parts (a) and (c) were answered well. However, Part (b) was poorly answered. Generally, candidates demonstrated little knowledge of ADT operations.

Question 13

This question assessed candidates' knowledge of the tools typically available in a programming environment. This question was generally poorly answered. Candidates were unable to identify or describe editor features. Some even confused this development tool with a newspaper editor. In Part (b), candidates were able to state the tools but could not describe them.

Question 14

This question examined candidates' knowledge and understanding of the activities of the implementation phase of software development. This question was fairly well done. Most candidates knew what code libraries were and the purpose of user documentation. Some candidates however did not describe the advantages of using code libraries during the coding process even though they knew what they were. Many candidates did not realize that the testing required was unit testing, system testing or dry-run testing.

Question 15

This question tested the candidates' ability to write code in an object-oriented programming language to implement a 'Square' class. Many candidates attempted this question, but the responses were generally very poor. The candidates had particular difficulty with the writing of constructors. In addition, many candidates did not know which methods did and did not require parameters.

PAPER 02

SECTION A – Software and Systems Development

Question 1

This question tested candidates' understanding of the software development life cycle models and the tools and techniques of the analysis phase of software development. The overall response to the question was for the most part satisfactory. There were many correct responses to Parts (b) (i) and (c). In Part (a) some candidate confused the word 'features' with 'steps' and so most of them simply listed the steps in the 'waterfall' approach. Parts (b) (ii), (b) (iii) and (d) were poorly answered. Even though it is clear that candidates are familiar with evolutionary development, they are unable to discuss the pros and cons of one development approach viz a viz another. The responses to Part (d) expose the need for more practice in the use of data flow models to document the flow of data.

Question 2

This question examined candidates' ability to differentiate between: 'top down' and 'bottom up' design strategies; functional and object-oriented design strategies; and to identify the features of good software design. In Part (c), an Entity-Relationship Diagram (ERD) was needed to capture given information. Candidate responses indicated that this was a less popular question but the responses were generally satisfactory. Some of the strong points in this question were the ability of the candidates to correctly draw the ERD, and their understanding of the differences among the various design strategies. The weak point was that many candidate responses to Part (b) (i) showed that they were unclear about what was required and consequently they were unable to answer Part (b) (ii). It seems this was a first exposure to this type of question.

SECTION B – Programming Languages

Question 3

This question tested candidates' understanding of the major stages in the compilation process and their ability to compare the procedural and object-oriented paradigms. Candidates were also required to write recursive and non-recursive functions that involved finding the factorial of a non-negative integer. Parts (a) and (b) were generally well done. However, in Part (a) the candidates did not explain the stages in sufficient detail. For Parts (c) and (d) few candidates were able to formulate a solution. The majority simply did not know how to approach the problem. More time has to be spent on the development of algorithms in the problem solving process.

Question 4

This question assessed candidates' ability to identify and use programming constructs, features and techniques in both the object-oriented and procedural paradigms. Candidates' strength was demonstrated in their ability to identify objects in an application, state, with type, instance variables for each object and use examples to differentiate between bounded and unbounded iteration.

Some candidates thought that the use of the word 'specify' in Part (d) meant that they were to write code. The responses to Part (g) indicated that the candidates were guessing at the role of the linker in a programming language. The candidates performed generally well in this question.

SECTION C – Program Development

Question 5

This question tested candidates' ability to develop an object-oriented application to keep track of course and student information for a university. Candidates' understanding of abstract data types (ADT) and their operations was also tested. This was a very popular question among candidates, but the responses were generally poor. The only strong point was the ability of candidates to declare the class attributes in Part (a) and to identify the appropriate ADT and operations in Part (b). Writing code for the class – constructor and methods proved very difficult as candidates did not understand what was required. In the classroom greater attention needs to be paid to the syntax of the language which the candidates are expected to use in the examination.

Question 6

This question tested candidates' knowledge of the operations of the queue abstract data type (ADT), and their ability to write code to implement the queue ADT as a class. Few candidates attempted this question. Those that did attempt it were able only to describe the function of each of the operations. Writing code proved extremely difficult for the candidates. For the most part there was a clear lack of understanding of object-oriented language programming. Candidates' performance on programming questions can be improved only with practice in the classroom.

PAPER 03

INTERNAL ASSESSMENT

GENERAL COMMENTS

Candidates performed well on the Internal Assessment. For the most part the projects areas were well chosen, covering a variety of areas and the reports were generally of good quality. Candidates did best when the projects were based on real situations which they had experienced.

SPECIFIC COMMENTS

There are some concerns which the moderators have with the samples submitted.

Unit 1

The main areas of concern are

- The awarding of marks for work not submitted. In one instance, the teacher awarded full marks for Networking when the candidate did not present any work on networking.
- The *Data Analysis* and *Organizational Impact* sections of the report were poorly represented in the project submitted.

Teachers (especially new teachers) are reminded that they should be fully familiar with the requirements and composition of the Internal Assessment and are to be guided by the criteria set down in the syllabus.

Unit 2

- Some of the samples submitted were not of the standard expected of candidates at the CAPE level.
- All section of the report must me submitted since moderators must be able to verify the marks awarded to each candidate.
- Testing of projects was often totally inadequate. Sample output from screens, must be included as part of the documentation to verify that the program is working.
- In many cases the user documentation was inadequate.
- The written report is expected to reflect the work done from the beginning of the project, not simply to concentrate on the finished product.

CARIBBEAN EXAMINATIONS COUNCIL

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATIONS MAY/JUNE 2005

COMPUTER SCIENCE

COMPUTER SCIENCE

CARIBBEAN ADVANCED PROFICIENCY EXAMINATIONS MAY/JUNE 2005

INTRODUCTION

This is the fifth year of open examinations for Unit 1 and the fourth year for Unit 2. There were three examination papers in both units, namely, Paper 01, Paper 02, and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the Internal Assessment, was examined internally by teachers and moderated by CXC.

In each unit, Paper 01 consisted of short-answer questions that were designed to test candidates' breadth of coverage of the syllabus. On the other hand, Paper 02 consisted of essay-type questions that were designed to test their depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

In each unit, Paper 01 consisted of three sections, each one corresponding to a module of the syllabus. There were five compulsory questions within each section, and each question carried 10 marks. The maximum amount of marks that could be obtained in Paper 01 was therefore 150.

Similarly, Paper 02 of each unit consisted of three sections, each one corresponding to a module of the syllabus. There were two questions within each section, and each question carried 30 marks. Candidates were expected to answer one question from each section. The maximum amount of marks that could be obtained in Paper 02 was therefore 90.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade remained 50 percent, 30 percent, and 20 percent, respectively.

GENERAL COMMENTS

In general, performance on both units has continued to improve, compared to previous years. Performance in Unit 1 suggests that teachers and candidates are completing the syllabus and are becoming more familiar with what is required.

Even though the performance in Unit 2 has improved, there is still concern about the level of programming ability being demonstrated in Sections B and C of both written papers. Candidates continue to find it extremely difficult to write even simple programs, especially in the object-oriented programming languages.

A major problem detected in the responses to questions in both Units that require the writing of prose is that of candidates' inability to clearly express their answers. Candidates need to understand what is required in a question and should learn to distinguish between key words such as “state”, “list”, “explain”, “discuss”, “describe” and “identify”. Many candidates lose marks by making a simple statement on a topic and not discussing the topic in sufficient detail. Other candidates also need to learn to label their responses appropriately and not lump together the answers to the different parts of a question in an essay. Since this is an advanced level examination, candidates are also advised of the importance of handwriting and the need for maturity in their responses.

Finally, candidates are encouraged, as part of their examination technique, to read questions carefully before answering, and to respond with sufficient detail that is commensurate with the marks indicated in the question.

DETAILED COMMENTS

UNIT 1

PAPER 01

SECTION A – Components of Computer Systems

Question 1

This question examined candidates' knowledge of server computers in a computer network and their ability to distinguish between two different types of cabling used in a computer network. Most candidates seemed to be quite knowledgeable of the role of a server in a client/server network and the question was well answered by most of them. In Part (a)(ii), candidates were generally unable to name specific types of servers typically found in a computer network (e.g., printer server and mail server). Some candidates mistakenly believed that peer-to-peer, client-server, hub, and routers were types of servers and went on to explain the purpose of these terms.

Question 2

This question tested candidates' understanding of operating system concepts such as memory protection, scheduling, and file management. It also required candidates to identify two applications in which a supercomputer can be used. The responses indicated that candidates had a very limited understanding of the concepts of memory protection and scheduling algorithms. Some candidates even wrote that memory protection refers to physically protecting the hard drive. However, the majority of candidates gave good responses for Parts (a) and (d).

Question 3

This question tested candidates' understanding of the data and processing requirements that should be considered before purchasing a computer system. This question was not done well by the majority of the candidates. Data requirements include considering the volume of data to be stored, sharing data, and securing data. Processing requirements include considering the speed and type of processing.

Question 4

This question examined candidates' knowledge and understanding of local area networks and different types of network configurations such as distributed, peer-to-peer, and centralised. The overall performance on this question was quite good, with an average of almost 6. However, a number of candidates confused the term 'configuration' with 'topology' and 'network type', and gave answers for network topologies and network types. In Part (b), while it was clear that candidates understood what is a local area network, their diagrams were badly drawn.

Question 5

This question tested candidates' knowledge and understanding of various concepts related to computer memory, such as 'volatility', 'memory size' and 'word size'. It also required candidates to describe one specialised feature of a mouse as an input device. However, it is felt that some candidates could have obtained more marks if they had not misinterpreted Part (a)(i) to be "What is volatile memory". 'Volatility of memory' refers to the degree that data in memory can be lost easily, for example, when electricity goes. Also, in Part (b), instead of describing a specialised feature of a mouse such as a scroll button, some candidates incorrectly gave examples of specialised input devices while others described the purpose of a mouse.

SECTION B – Application of Computers

Question 6

This question tested candidates' understanding of computer applications in an organization as well as the implications of computer automation. Most candidates were able to outline at least one benefit and one negative consequence of using computers in an organization.

Question 7

This question tested candidates' knowledge and understanding of various types of information systems such as transaction processing systems (TPSs) and data warehouses. Candidates were also required to discuss what happens during data mining. As with similar questions in the past, most candidates answered the question poorly and the average mark was less than 4. Some candidates were unable to give an actual example of a TPS though they correctly named the inputs and outputs of the system. Others had difficulty in distinguishing between the inputs/outputs of a system and the input/output devices for that system. Using automatic teller machines as an example, some candidates incorrectly identified keyboard/mouse and printer/monitor as inputs/outputs of the system, respectively. The responses for Part (b) were generally vague, demonstrating candidates' limited knowledge of data warehouses and data mining. For example, a common response was that a data warehouse is a location where data is stored.

Question 8

This question required candidates to demonstrate their understanding of special-purpose computer systems, specifically, expert systems and embedded systems. Many candidates gave good examples of expert systems and correctly identified the inputs/outputs of these systems, although some had problems identifying inputs/ outputs as explained in Question 7 above. However, the inference engine was generally omitted in describing an expert system in Part (a). Some candidates also confused expert systems with embedded systems. Many candidates incorrectly stated that an embedded system is a piece of software that is added to another larger piece of software such as an operating system. Candidates were expected to state that an embedded system is a computer system placed inside other products to add features and capabilities that are computer-based (e.g., an embedded system is used inside a microwave oven).

Question 9

This question tested candidates' understanding of the concepts of telecommuting and videoconferencing and was answered fairly well. Most candidates were able to correctly explain how communication is achieved in videoconferencing. However, many candidates confused telecommuting with teleconferencing and telemarketing. Telecommuting refers to a work arrangement where employees work away from their offices using personal computers (e.g., at home), communicating with their offices using email and other forms of communication technology. The advantages of telecommuting and videoconferencing were also not clearly stated by many candidates.

Question 10

This question tested candidates' understanding of data security and how to protect data in an organisation, both from damage and from unauthorised access. In Part (a), most candidates identified two ways in which data could be accidentally damaged but did not state exactly what damage was caused. However, in Part (b), most candidates were knowledgeable about the techniques that could be used to protect data from unauthorised access.

SECTION C – Computer-based Problem Solving

Question 11

This question examined candidates' knowledge of using different computer-based tools to solve the real-life problems of generating a report on the performance of candidates in the CSEC Information Technology examinations and maintaining student information at a school on a day-to-day basis. This question was well answered by most candidates. . In both Part (a) and Part (b), candidates were able to correctly identify appropriate software tools (word processor and spreadsheet, and database management systems, respectively). However, several candidates did not clearly state the features of these tools that made them appropriate for the task at hand. As indicated in the question, it is important for candidates to link the features of the tools to the task that must be performed.

Question 12

This question examined candidates' knowledge of using a software tool to solve the real-life problem of making a presentation of a company's sales performance. Candidates were also expected to understand the kinds of problems that could affect the use of the tool selected and how to solve these problems. This was a fairly easy question and it is surprising that candidates' performance was not much better than it was. The obvious software tool for the problem was presentation software such as Microsoft Powerpoint. However, some candidates stated that spreadsheet software could be used and this was also accepted. Similar to Question 11, candidates were often unable to state three features of the tool that made it appropriate for the task. Several candidates gave simplistic responses for Part (b) such as the software and presentation data must be stored on a computer. It should be noted that the questions in this section deal with Computer-based Problem Solving, so the need for computers, software, and data is clearly understood. More appropriate responses on potential problems that can occur when using the software tool are that the vice-president may not know how to use the software (i.e., training is required) or that the presentation file may become damaged for some reason (requiring backup measures to be put in place).

Question 13

This question tested candidates' knowledge and understanding of information sources and their ability to select appropriate knowledge sources for a given information need. The question was well answered by the majority of the candidates. Several candidates obtained full marks in this question. However, given its straightforward nature, a higher average was expected.

Question 14

This question tested candidates' knowledge and understanding of the characteristics of information (with an emphasis on reliability) as well as their ability to analyse a situation to determine the quality of information that can be obtained. It was generally answered satisfactorily with many candidates losing marks for not developing their responses or repeating a source of un/reliable information already given. Also, the majority of candidates were unable to furnish an example in Part (a) (i) when they explained the term 'reliability of information'. In Part (b), the reasons given for not accepting the information were often quite poor or underdeveloped. Candidates were expected to give responses such as the receptionist had no authority to give out the information requested and may also give inaccurate information since this was not his/her job area.

Question 15

This question tested candidates' knowledge and understanding of a digital library as an information source. It was clear that many candidates had limited understanding of a digital library, and many of them opted to answer the question from the point of view of a CD. A digital library is a database containing documents such as journal and magazine articles in digital form that can be accessed by users and downloaded to their own computers. Candidates also gave a general list of advantages and disadvantages of a digital library, without attempting to elaborate on the points stated. This may be attributed to their limited understanding of a digital library.

PAPER 02

SECTION A – Components of Computer Systems

Question 1

This question tested candidates' in-depth knowledge and understanding of a number of hardware-related and network-related concepts covered in the unit. Generally, candidates were able to differentiate between ROM and EPROM, MAN and LAN, and between workstation and supercomputer. However, some displayed weaknesses in comparing a hub and router and in comparing EPROM and EEPROM. The characteristics of the transmission media given were also well described. However, weak responses were obtained for Part (c), when describing network-related problems that can be solved using diagnostic tools. Many candidates also seemed unfamiliar with compression utilities. Their discussion of radio waves as a transmission medium was often poor.

Question 2

This question tested candidates' understanding of issues related to purchasing a mouse, the role of the user interface in an operating system, and various network-related concepts such as microwave transmission and the FDDI and ring network topologies. Candidates gave very good responses on the issues related to purchasing a mouse, but the role of the user interface in an operating system was not clearly explained. Also, most candidates were unaware that FDDI is composed of two rings of fibre-optic cable. This caused them to lose marks in the other sections of Part (d). Candidates generally gave good answers for how data is transmitted in a ring network, but often failed to mention the presence and purpose of the token.

SECTION B – Application of Computers

Question 3

This question tested candidates' understanding of the changing information needs of an organization over time as well as their understanding of the historical changes in computer applications over time. Very few candidates (less than 5%) chose to answer this question. From the responses obtained, it seemed that few candidates understood what was being asked by the question and their responses were vague and off the point. Even though candidates were able to identify changes in storage and processing capabilities, they did not relate these changes to the specific needs of the business, such as handling an increased number of customers, employees, products, etc. In Part (c)(i), candidates were generally unable to specify the information processing activities that would take place at the operational and strategic levels of the organization. Candidates need to be able to better distinguish between the operational and strategic aspects of information processing in an organization. However, it is encouraging to note that in Part (c)(ii), several candidates were able to correctly identify transaction processing systems, management information systems, and other systems that could be used at the organization.

Question 4

This question tested candidates' in-depth understanding of data security and how to protect data in an organisation, both from unforeseen circumstances such as hurricanes and fire, and from unauthorised access. More than 95% of the candidates attempted this question and all parts of this question were generally answered well. In particular, there were very good answers for Part (a)(i) when candidates were asked to discuss two means by which unauthorised access to data could occur in the bank.

SECTION C – Computer-based Problem Solving

Question 5

This question tested candidates' knowledge and understanding of using different computer-based tools to solve a real-life problem of attracting investors to a company. Only about 30% of the candidates chose to answer this question instead of Question 6. Candidates were able to correctly identify software tools and explain how these could be used in the company's venture to attract investors. However, marks were often lost because an appropriate discussion was not given.

Question 6

This question tested candidates' knowledge and understanding of information sources and evaluative criteria that can be used to determine the validity of an information source. It also tested candidates' understanding of the problem solving process required for developing a computer-based solution for a business problem. About 70% of the candidates opted to answer this question. Part (a)(i) was answered well by most candidates who correctly identified four other sources of information that could have been used to confirm the information obtained. Part (b) was also answered well by most candidates who recognised that many different kinds of information could be obtained from the Internet. However, candidates were weak in their understanding of evaluative criteria for information, such as currency and authorship. Weak responses were also obtained for Part (c) when describing the four stages of the problem solving process.

UNIT 2

PAPER 01

Section A – Software and Software Development

Question 1

Candidates answered the question generally well although many of them did not fully explain their answers. Most candidates had some concept of the answers required and most stated responses either exactly as required or provided synonyms for the terms required. A few candidates interpreted the term ‘development’ as ’design’.

Question 2

This question was poorly answered. One strong point was that candidates knew the steps for the waterfall model. Some candidates did not understand what was required

for Part (a) and simply stated the phases. Generally, candidates had a poor understanding of the waterfall model when compared to the evolutionary model of software development. Those who understood the models emphasized the wrong points. For Part (c), some candidates needed to understand that stating an advantage/ disadvantage is not enough; it must also be explained.

Question 3

This question was answered fairly well. There were many good responses for Part (a). Some candidates confused the feasibility study with the steps of the development process.

Question 4

This question was answered fairly well by most of the candidates. Generally, the data flow diagrams were well done. However, misconceptions on functional/non-functional requirements were present.

Question 5

This question was generally done satisfactorily. Candidates had a strong understanding of user-interface design principles. There were some misinterpretation and confusing arguments in some of the responses of the candidates.

Section B – Programming Languages

Question 6

There was a **major** misinterpretation of this question by candidates. Some candidates focused on the systems development life cycle rather than the characteristics of a program. Generally, the candidates who scored between 6 to 8 (highest) had a good knowledge of the concepts and theory being examined. A few candidates misinterpreted the question as the "stages of a compiled program" and the "software life cycle".

Question 7

Generally, the performance on this question was good and there were several cases where candidates got full marks. Most candidates knew how to describe imperative languages. However, many candidates were unable to describe declarative languages. For Part (b), some candidates thought that defining a top-down approach was specifying an advantage.

Question 8

Overall responses for this question were very good but in some cases, groups of candidates performed very poorly. In many cases candidates achieved full marks. Candidates did very well at distinguishing between 2GLs and 3GLs. Many candidates did not relate their explanations of the factors given (such as runtime / portability / maintainability) to the issue of choice of programming language. In the context of choice of programming language, the responses were generally vague. Many candidates could explain the factors given (in isolation), but displayed confusion when attempting to say how the factors influenced the choice of a programming language.

Question 9

The performance on this question was fair. Logical thinking by candidate was generally not displayed. Loops were in the wrong place and many candidates had the correct logic, but lost marks when printing the required value. The drawing of flow charts instead of writing an algorithm was one of the weak points. The use of an array to store the values instead of the use of two variables would have also been a correct approach to this question. Candidates need to read the question to determine what is required. More practice on the logical flow of a solution should be done. Practical coding would benefit both candidates and teachers.

Question 10

This question was not done very well. Many candidates were able to define the term algorithm. However, some candidates could not distinguish between bounded and unbounded iteration and it was clear that they were not familiar with the terms. Also, many candidates did not have a good grasp of classes and objects as well as private and public methods.

SECTION C – Program Development

Question 11

The responses in this question were fairly good. Part (a) and Part (b)(i) were answered fairly well. However, Part (c) was challenging to candidates. Many candidates were able to identify the user interface objects and describe the purpose of these objects quite well except for fonts. Most identified the event in Part (b)(i) correctly. In Part (b)(ii) many candidates wrote algorithms based on procedural programming. Some candidates actually wrote programming code for Part (b)(ii), but an algorithm was required. Future candidates are encouraged to gain practice in writing algorithms for event-driven programming problems.

Question 12

This question was not well done by most candidates. The candidates who knew the material scored higher marks and the remainder performed badly. Some candidates drew the stack and showed the contents or used trace tables. Some candidates did not show the contents of the stack during each pass of the loop and just wrote the output. Many candidates did not seem to understand the purpose of the "else" statement. In the end, many could not figure out the purpose of the algorithm. Future candidates and teachers are encouraged to gain practice in tracing and understanding algorithms by using more practical examples of ADTs.

Question 13

The responses to this question were generally not of a high quality. In many cases, it was clear that candidates did not know the proper syntax for programming statements and tended to give answers in pseudocode. Candidates need to obtain more practical experience in programming to answer the questions in this section.

Question 14

The responses to this question were generally good. The strong point of this question was candidates' ability to correctly list four tools typically provided in a programming environment. However, the explanation of how these tools are used was often poor, with few candidates being able to differentiate between a tracer and a stepper. DFDs, HIPO charts etc. were common responses. It is suggested that candidates make more active use of the tools in a programming environment so that they can answer questions of this nature.

Question 15

The responses to this question were fairly good. Most candidates were able to list two types of documentation and describe them properly. More than 50% were able to get at least 1 out of 2 for Part (a)(i). Some candidates gave the same explanation for unit testing and system testing. The key difference between unit and system testing needs to be understood by candidates. A good response should have included a phrase like "integration of modules" when describing system testing. More practical examples should be used in demonstrating to candidates how system testing and unit testing is done.

PAPER 02

Section A – Software and Software Development

Question 1

The question was fairly well answered by candidates. Candidates comfortable with data flow diagrams (DFDs) gained most of the marks for this question. Most candidates were able to identify the customer entity in the DFD. Some candidates were able to explain what is an entity-relationship diagram (ERD) but were unable to differentiate between an ERD and a data dictionary. Candidates often confused data flow diagrams with ERDs and flow charts. Hence they were unable to give good answers for Part (b). Candidates were also unable to give a good example to explain what is a data dictionary. Future candidates need to gain more practice in drawing DFDs. The distinction between DFDs and other types of diagrams such as ERDs should also be carefully made and their respective symbols clearly distinguished.

Question 2

In Part (a), most candidates were unable to properly explain the use of CASE Tools. Part (c) of the question was answered well by many candidates. The strong points of candidates in this question were in identifying the attributes and relationships in Part (b), defining object-oriented design in Part (c), and listing the design process activities in Part (c). . Candidates also need to know the proper symbols to use when drawing an entity-relationship diagram (ERD) and the features of function-oriented design. Future candidates also need to gain more practice in drawing ERDs and understanding the differences between other types of diagrams such as data flow diagrams.

Section B – Programming Languages

Question 3

Candidates demonstrated a poor grasp of several basic concepts in computer programming and the only strong point observed was in their explanations for Part (a), on generations of programming languages. Most candidates had no understanding of the concept of recursion and how to write a recursive algorithm. Candidates should spend some more time on the topic of recursion and in understanding control structures (if-else, loops).

Question 4

The responses to this question were generally good. The concept of an object, instance variables, and classes were understood by most of the candidates. Parts (b) and (c) were generally answered well; however, Part (c) had some poor answers. In Part (a), some candidates were unable to identify the accessor and modification methods required, so they just wrote any methods that came to mind. Candidates also demonstrated weakness in explaining the concept of encapsulation. In Part (a)(iv), they also had some problems in drawing the class diagram for the *Plane* object. In the future candidates should practice drawing class diagrams listing the attributes, and methods of a class.

SECTION C – Program Development

Question 5

This question was poorly answered, especially where ADT operations were required. The best-answered portion of the question was Part (a), which required candidates to identify different types of GUI controls for the user interface. Candidates need to develop application and synthesis skills to tackle the problems in this section. Many candidates are weak at problem solving and algorithm design. To perform well in this section, future candidates need to improve in these areas.

Question 6

This question was generally not answered well. However, there seemed to be an improvement in the quality of responses to this type of question, compared to the examinations of 2004. Some candidates are still not able to write proper programming statements. In Part (a), many candidates did not understand what "constructors" and "accessors" were. In Part (b), the "addStudent" method was generally poorly done, with incorrect code being written. Part (c) was generally well done except for section (iii), which involved adding *Student* objects to a *Faculty* object. This operation required the *Faculty* object to have a data structure such as an array that could be use to hold the *Student* objects. The "addStudent" method is then implemented by inserting the *Student* object into the array at an appropriate position (e.g., the end of the array).

PAPER 03

INTERNAL ASSESSMENT

GENERAL COMMENTS

The performance on the Internal Assessment was generally good. However, compared to the performance in 2004, there was a decline in both units, especially in Unit 2. It was observed that the projects submitted by candidates were deficient in various aspects and were still being awarded high marks by teachers. There were also a number of cases where it was clear that candidates did not know exactly what was required of them in the Internal Assessment and consequently, obtained very low marks. Better performances on the Internal Assessment should lead to better overall performances on both units as well as better performances on the other papers. Candidates need to maximize the opportunity to get higher marks on the Internal Assessment. Teachers also need to become more closely involved in the supervision of the projects. This should lead to an improvement in candidates' performance on the Internal Assessment, solidifying their understanding of the theoretical aspects of the units, which in turn, will lead to an overall improvement in performance.

Generally, most candidates chose appropriate topics for the Internal Assessment. The topics chosen were relevant to the level of the candidates and the specific objectives of the respective syllabuses. The treatment of the topics by the candidates was adequate. A small percentage was comprehensive though some tended to be superficial. The reports were also generally well presented and teachers complied with requirements such as ensuring that there was a cover page for each project and entering the marks on the required form.

DETAILED COMMENTS

Unit 1

Some of the projects submitted demonstrated below average performance in a number of areas such as description of context, purpose of study and solution process. In many cases, the responses to these areas and others defined in the syllabus were non-existent, vague, or poorly expressed. Often, marks were awarded to areas that were not specified in the syllabus and in those instances teachers were not adhering to the mark scheme provided. In other cases, full marks were awarded even though the defined criteria were completely absent.

In the problem identification section, it was clear that a number of candidates did not understand the term “external entities and processes”. The description of the current system and procedures also needed improvement. Candidates took the purpose of their study straight from items described in the syllabus. However, these items need to be tailored to their specific projects. Finally, candidates need to be reminded that there is a word limit for their projects.

Unit 2

There were a few cases where candidates used the old Unit 2 syllabus to prepare their Internal Assessment. As a result, irrelevant information was submitted. Similar to Unit 1, marks were often awarded in a manner that was not consistent with what is defined in the syllabus.

From the moderation of the Internal Assessment, it is clear that many candidates do not understand how to write programs in a programming language. In some cases, candidates wrote pseudocode where program code was required and this was awarded full marks. In most cases, the functionality of the program written was poorly described and there were no screen shots of the working system displayed in the reports (perhaps indicating the inability of the program to work in the first place).

CARIBBEAN EXAMINATIONS COUNCIL

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION MAY/JUNE 2006

COMPUTER SCIENCE

COMPUTER SCIENCE

CARIBBEAN ADVANCED PROFICIENCY EXAMINATION

MAY/JUNE 2006

INTRODUCTION

This is the sixth year of open examinations for Unit 1 and the fifth year for Unit 2. There were three examination papers in both units, namely, Paper 01, Paper 02, and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the Internal Assessment, was examined internally by teachers and moderated by CXC.

In each unit, Paper 01 consisted of short-answer questions that were designed to test candidates' breadth of coverage of the syllabus. On the other hand, Paper 02 consisted of essay-type questions that were designed to test their depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

In each unit, Paper 01 consisted of three sections, each one corresponding to a module of the syllabus. There were five compulsory questions within each section, and each question carried 10 marks. The maximum amount of marks that could be obtained in Paper 01 was therefore 150.

Similarly, Paper 02 of each unit consisted of three sections, each one corresponding to a module of the syllabus. There were two questions within each section, and each question carried 30 marks. Candidates were expected to answer one question from each section. The maximum amount of marks that could be obtained in Paper 02 was therefore 90.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade remained 50 per cent, 30 per cent, and 20 per cent, respectively.

GENERAL COMMENTS

In general, performance on both units of the syllabus has continued to improve. In Unit 1, more than half of the candidates are now obtaining grades I to III. In Unit 2, more than 40 per cent of the candidates are now obtaining grades I to III. However, the performance on the School Based Assessment should have been better. Better performances on the SBA should lead to better overall performances on both units as well as better performances on the theory papers. Candidates need to maximize the opportunity to get higher marks on the SBA.

Even though the performance in Unit 2 has improved, there is still concern about the level of programming ability being demonstrated in Sections B and C of both written papers. Candidates continue to find it extremely difficult to write even simple programs, in both the imperative and object-oriented programming languages. Teachers are encouraged to have several programming labs and exercises done with the candidates.

As recommended last year, candidates are encouraged, as part of their examination technique, to read questions carefully before answering, and to respond with sufficient detail that is commensurate with the marks indicated in the question.

GENERAL COMMENTS

UNIT 1

In general, performance on both units of the syllabus has continued to improve. In Unit 1, more than half of the candidates are now obtaining grades I to III. In Unit 2, more than 40 per cent of the candidates are now obtaining grades I to III. However, the performance on the School Based Assessment should have been better. Better performances on the SBA should lead to better overall performances on both units as well as better performances on the theory papers. Candidates need to maximize the opportunity to get higher marks on the SBA.

Even though the performance in Unit 2 has improved, there is still concern about the level of programming ability being demonstrated in Sections B and C of both written papers. Candidates continue to find it extremely difficult to write even simple programs, in both the imperative and object-oriented programming languages. Teachers are encouraged to have several programming labs and exercises done with the candidates.

As recommended last year, candidates are encouraged, as part of their examination technique, to read questions carefully before answering, and to respond with sufficient detail that is commensurate with the marks indicated in the question.

DETAILED COMMENTS

UNIT 1

PAPER 01

SECTION A – Components of Computer Systems

Question 1

This question tested how a UPS could help protect a computer (part (a)), the concept of port connectivity (part (b)) and the use of satellites as a transmission medium (part (c)). Several candidates obtained full marks in this question. However, many candidates ignored the marks available for part (a) and just wrote one point. A good response for part (a) would have mentioned that a UPS gives users time to close applications and save their work. Part (b) was generally well answered with many candidates saying that port connectivity had to do with advances in port design and cable types. For part (c), some candidates failed to mention that satellite signals propagated to space and then to earth.

Question 2

This question tested candidates' knowledge of the instruction set of a computer, client/server software and peer-to-peer networks. Some candidates thought that the instruction set was the programming language statements of a particular language e.g. Pascal. For part b(iii), some candidates did not know that the client makes a request which the server attempts to service in the specific applications chosen. For part 2(a), a good response was 'the instruction set of a computer refers to the complete set of instructions which are hard-wired into the micro-processor and which defines the set of instructions the computer will understand and be able to execute'.

Question 3

Question 3 looked at features of disk management software, memory management in an operating system, multitasking and multiprocessing. Responses for this question were generally good and many candidates scored full marks. Weaker candidates mixed up 'multitasking' and 'multiprocessing' in part (c).

Question 4

This question required the candidates to know about features provided by network operating system software as well as two utilities typically present in system software. The question attracted some of the best responses in the examination. Candidates displayed a sound knowledge of most features provided by network operating system software. Weaker candidates could explain either none or only one of the utilities typically present in system software.

Question 5

In this question, candidates had to compare extranets to the Internet, explain the term 'capacity of a storage device', explain 'access speed of a storage device' and explain 'portability'. Except for parts a(i) and a(ii), most of the other parts were generally handled well. In part (a), some candidates confused 'intranet' with 'extranet'. Part b(i) which referred to the capacity of a storage device was well answered, with some candidates giving values of actual device capacities e.g. 100 MB.

SECTION B – Application of Computers

Question 6

In this question, candidates had to demonstrate knowledge of expert systems, data warehouses and data mining. This question was generally poorly done. In Part (a), most candidates did not know what an expert system was. An expert system stores knowledge of a limited domain in the form of rules and makes inferences based on those rules, similar to a human expert. Also, many could not describe data warehouses. However, Part b(ii) was answered better than Part (b)(i).

Question 7

This question examined monitoring and control systems, user interfaces and suitability of a computer application for an organization. Most candidates were not familiar with monitoring and control systems as tested in Part (a). Monitoring and control systems use computer technology to track/monitor processes and take certain courses of action if problems occur. They are typically present in many industrial applications. Parts (b) and (c) were well-answered.

Question 8

This question tested candidates' knowledge of data security, system back-up and real time updating of files. Most candidates answered Part (a) fairly well; however some did not describe, but rather just listed, the main points. There are logical and physical approaches to data security. In the logical approach, programs can have password protected access. In the physical approach, computers can be placed in enclosed/protected areas. In Part (b), many candidates explained system back up but did not give an example. An example could be a secretary backing-up her word-processing files on a file server at the end of the day. Part (c) was done well by most candidates.

Question 9

This question started by asking how firewalls and encryption could protect an organization's data from unauthorized access. The other parts required a suggestion of ways in which computers could affect a financial organization and the distinctions between batch and on-line processing. Responses for this question were generally of a poor quality although some candidates scored full marks. Part (a) was generally answered fairly well. One candidate stated that firewalls 'helped protect an organization's data from unauthorized access'. However, this was already stated in the question and gained no marks. Part (c) contained important types of processing used in information technology but many did not know the meaning of the terms. Batch processing is where transactions are gathered together into a collection and are used to update master files at a later time. Online processing is where transactions are executed and changes are made to master files almost immediately.

Question 10

This question examined telecommuting and the ways in which computers help us entertain ourselves and facilitate communication. In Part (a), many candidates mixed up telecommuting and teleconferencing. Telecommuting is where employees can work from one location and communicate their work to the main office at another location using computer and telecommunication technologies. Parts (b) and (c) were well-answered.

SECTION C – Computer-based Problem Solving

Question 11

Part (a) of this question asked for similarities and differences between information stored on CD-ROM and in a digital library. Part (b) pertained to sources of information relevant to monitoring the progress of a hurricane. Part (a) was generally well answered with many candidates scoring full marks. For Part b(i), approximately half of the responses stated that newspapers were a good source due to the fact that it was updated regularly. However, it should be noted that newspapers are not updated regularly enough to monitor a hurricane. Parts (b)(ii) and b(iii) were generally well answered.

Question 12

This question was about information and characteristics of information sources related to making a financial decision. The businessman got his initial information from an anonymous email. Part (a) was not answered very well as candidates did not incorporate certain key words in their responses which should have related to information characteristics.
For Part (b), many candidates *listed* sources rather than *listing* and *explaining* them.

Question 13

This question looked at data sources and their characteristics. This question was a fairly simple question, but many candidates answered it poorly. Questionnaires were a good source of information in this case. Part (b) attracted some fairly good responses.

Question 14

For this question, candidates had to distinguish between DVD and CD ROMs as information sources. Part (b) asked about sources of information based on a scenario. Overall, this question was answered well and some candidates scored full marks. Part (a) was well answered. In Part (b)(i), some candidates used sources that were not relevant to Francis' situation. Three good sources of information for Francis were newspapers, magazines and the Internet.

Question 15

This question was based on software tools appropriate for different tasks performed by a toy store at Christmas. For Part (a), most candidates did not give a very suitable response and hardly anyone obtained full marks. Spreadsheet was a good response for Part (a). In Part (b), most candidates were able to identify the software tool, but there were hardly any reasonable responses for precautions which should be taken. One precaution was to keep back-up hard-copies of the presentation.

PAPER 02

SECTION A – Components of Computer Systems

Question 1

Part (a) required candidates to differentiate between several pairs of computer terms. Part (b) examined how an operating system could execute three processes at the same time. Part (c) tested candidates' knowledge of the data requirements and communication requirements of a package.

a) Generally, candidates did not differentiate between the two terms given. Many candidates defined each term separately without highlighting what made them different. For Part (a) (iii), a good response was 'cache memory is much faster than RAM and also more expensive'.

b) Clearly, most candidates knew that the operating system could handle and execute many processes at the same time. However, only brief explanations of how this was done were given by most candidates. A good response should have contained a discussion of round-robin allocation of time-slices, preemption of processes, interrupts and saving and reloading of process states.

c) Some candidates seemed confused in Part (i) and spoke about qualities of information. They did not read the scenario carefully and answer based on the situation presented. Many also spoke about "database attributes". More focus should have been placed on the fact that different departments were involved.

Question 2

Part (a) of this question presented a diagram and candidates were asked to identify the network topology illustrated. Part (b) asked for two advantages of fibre-optic cable over coaxial cable, Part (c) asked about the OSI model for computer communication and Part (d) looked at data transmission in bus and ring networks. Overall, this question was answered very well in most instances. For Part (c), some candidates did not focus on the key functions of the various layers of the OSI model. For example, the main role of the network layer is to route packets although there are other subordinate functions.

SECTION B – Application of Computers

Question 3

Part (a) required candidates to use examples to distinguish between data loss and data corruption. Part (b) examined levels of management and the associated information systems. Part (c) asked about precautions to minimize data loss in the event of a hurricane. Overall, the question was generally well answered although it was not a popular question. In Part (b)(ii), many candidates knew of the different types of information systems but had problems relating them to the levels of decision making. Parts (a), (b)(i) and b(iii) were fairly well done as were Parts 3(c)(i) and 3(c)(ii). Some weak responses for 3(c)(ii) were UPS, surge protectors and covering computers with plastic bags. One precaution was 'to move data to a safe storage location in the building: the safe storage location could be a sturdy cabinet that was waterproof and located in a sturdy room'.

Question 4

This question looked at techniques to deter unauthorized access to a web based system, software piracy, ways in which an organization could benefit from automation and three ways in which computer applications have affected the job market. The question was very popular. Parts (a) and (b) were satisfactorily done.

Many responses for Parts (c) and (d) were similar. For Part (c), some responses were too simple and involved expressions like "more work done", "less salaries to pay out" etcetera. There was a clear indication that some candidates misinterpreted Part (d). Also, positive and negative ways in which computers affect the job market should have been considered for

Part (d). For example, new jobs have been created, for example,. software developer and computer applications have caused displacement of a number of clerical employees in some organizations.

SECTION C – Computer-based Problem Solving

Question 5

Part (a) of this question presented a scenario and candidates had to describe three other sources of information to confirm information found on a web site. A discussion of evaluative criteria for acceptance or rejection of information was then required. Part (b) asked for the benefits to an organization of using the Internet as an information source and part (c) required a discussion of three characteristics of information sources. Part a(i) was fairly well answered. Good responses for part a(i) included valid information from other related sources, Web sites, experts in the financial sectors, articles in reviewed magazines, newspaper articles and television programmes. Part a(ii) was badly done. Very few candidates were able to identify all three evaluative criteria correctly. Most candidates did not correctly relate their explanations to the stated criteria. Many candidates did not focus on “using the Internet as an information source”, but rather discussed the use of the Internet as a way to promote their business. Candidates should pay attention to what the question is specifically asking and read questions carefully. Part (c) was not well done. Many candidates were not clear about the definition for each term and hence were not able to obtain all three marks for each concept.

Question 6

The majority of the candidates attempted this question which was based on a University scenario that required the use of software tools for different tasks. Part a(i) was generally well-answered. A minority of candidates did not obtain the mark due to lack of familiarity with the names of appropriate software tools. For example, some cited Microsoft ® Excel as an example of a database software tool. Part a(ii) was fairly well answered although many candidates failed to explain their answers sufficiently in order to obtain full marks. Candidates need to pay attention to specific technical terms as it relates to the particular software tool. Parts b(i) (ii) had similar problems to Parts a (i) and a (ii). In Part (c), a minority of candidates obtained all six marks. Most candidates did not include in their response, the ‘retrieval of information from the database and spread sheet programs’, into the presentation. Many of them simply stated that Microsoft ® PowerPoint could be used. Part (d) was not well-answered. Many candidates cited word-processors as opposed to desktop publishing software as the most appropriate tool for producing the newsletter and brochures. Many of them were not able to effectively explain the reasons for their selection.

UNIT 2

PAPER 01

Section A – Software and Software Development

Question 1

This question tested candidates’ knowledge and understanding of the waterfall model, which is one of the most widely known models of a software development process. There were good answers to Part (a). However, many candidates did not properly explain that a software development process is a set of activities and associated results which lead to the production of a software product, saying instead that it is a sequence of steps. In Part (b), some candidates did not specify the names of the stages of the waterfall model or specified the names incorrectly. Poor explanations were sometimes given for the activities that take place in the three stages, especially for the design stage. Part (c) was generally answered well, though some candidates indicated that a weakness of the waterfall model is that it is time consuming and costly, instead of mentioning the inflexibility inherent in the model.

Question 2

This question tested candidates’ knowledge and understanding of two information gathering techniques that can be used during requirements analysis, namely, interviews and questionnaires. It also required candidates to determine which one was more appropriate for the scenario given. There were good answers to all three parts of this question and valid reasons were given for choosing one technique over the other in the situation given. However, there were cases where candidates stated that interviews allowed one to have face-to-face or direct encounter with stakeholders but did not indicate why this was valuable to the information gathering process. Some candidates also failed to state that responses to interview questions are documented and that responses to questions in a questionnaire are collected and analyzed.

Question 3

This question tested candidates' knowledge and understanding of Computer Aided Software Engineering (CASE) tools. The question was generally answered well by candidates though some did not answer the question at all. However, in Part (a), some candidates failed to mention that CASE tools are software products. Some failed to mention that it is used to support or automate different aspects of the software development process. In Part (b), some candidates did not identify the stages of the software process where CASE tools can be used. Few candidates mentioned pertinent terms such as "graphical system models" and "technical and user documentation". In Part (c), some candidates incorrectly stated that one disadvantage of a CASE tool is that software developers would become less skilled by using the tool.

Question 4

This question tested candidates' knowledge and understanding of architectural design which is an important design process activity. This question was poorly answered by most candidates, indicating inadequate preparation or understanding of this aspect of the syllabus. In Part (a), few candidates were able to explain that system structuring in the breaking down of the system into subsystems and some even confused this term with the software development life cycle. In Part (b), many candidates stated that modular decomposition is the breaking down of the system. However, this is partially correct; modular decomposition is the decomposition of sub-systems into modules. In Part (c), candidates were unable to distinguish between a sub-system and a module. Some of them incorrectly stated that a sub-system performs a single task. The responses to Part (d) were generally poor and candidates failed to mention that outputs include graphical representations of system models and diagrams showing the system structuring and modular decomposition.

Question 5

This question tested candidates' knowledge of user interface design guidelines and guidelines for designing reports. There were fairly good responses to Part (a) of this question. However, guidelines such as "user familiarity", "consistency", and "user guidance and feedback" were rarely mentioned. Also, instead of stating "recoverability" as the ability to gracefully recover from errors, candidates used a less appropriate term, "robustness". The responses to Part (b) were not satisfactory. Guidelines for report design such as using meaningful titles, presenting relevant information, and balancing the layout of the information were generally not described. Instead, candidates described the information that the reports should contain and how they should be prepared.

Section B – Programming Languages

Question 6

This question tested candidates' understanding of various object-oriented concepts and their ability to represent a class in a diagram. Candidates were required to explain the concept of inheritance and to describe how message passing takes place among objects. Part (a) was answered well by most candidates though data types for the instance variables and the "()" after method names were often omitted. In Part (b), candidates seemed to understand how message passing took place but did not clearly explain the process. Some incorrectly indicated that the *Customer* object would call the *Account* object using its *withdraw()* method as follows: *A.withdraw()*. Some candidates missed the point of the question completely and talked about the physical process of going to the bank to withdraw/deposit money using ATMs and bank slips. Part (c) was not answered well by most candidates and no response gained full marks. It should be noted that inheritance is when a class assumes the instance variables and methods of another class, perhaps adding its own instance variables and methods.

Question 7

This question tested candidates' knowledge and understanding of several fundamental object-oriented concepts. The answers to Part (a) were generally correct though some of the candidates were unable to properly distinguish between an object and a class. A class is like a template that describes the behaviour and attributes of a set of similar objects while an object is an instance of a particular class. Part (b) was not answered well and many candidates did not explain that encapsulation is the technique of hiding the internal implementation details of an object from external objects. For Part (c) few candidates explained that public instance variables can be accessed directly by external objects. Thus, if these variables change in the future, all the external objects would have to be changed since they were not shielded from the changes by encapsulation.

Question 8

This question tested candidates' knowledge and understanding of programming languages and the use of compilers and interpreters. Part (a) was answered well and most candidates were able to classify programming languages in terms of level such as high-level versus low-level. Some gave examples in cases where they could not identify specific terms and these were accepted. In Part (b), most candidates only gained half the marks since they only stated one reason, namely that compiled code executes faster than interpreted code. Another reason is that compiled code can be distributed and executed without need for the compiler, unlike interpreted code which always requires the presence of the interpreter. Part (c) was generally answered well though some candidates lost marks by missing out one or more criteria for selecting a programming language.

Question 9

This question tested candidates' ability to trace through an algorithm and to determine the output that it generates. The majority of candidates gained only partial marks for this question since they only had part of the output correct. Many candidates had no idea how to trace through the algorithm and simply rewrote the algorithm in their responses. However, a few candidates gained full marks by correctly drawing the shape that would be generated by the algorithm. The following is the expected output of the algorithm:

```
**********
*        *
*        *
*        *
**********
```

Please note that full marks were awarded for alternative output based on a different valid interpretation of one of the functions given.

Question 10

This question tested candidates' ability to write algorithms using bounded and unbounded iteration. Part (a) was answered well by many candidates. However, many candidates only obtained partial marks for Part (b). Several candidates attempted to use the modulus function but used it incorrectly. Generally, the responses by candidates in this question indicate that they are unable to write simple algorithms using the basic programming control structures of sequence, selection, and repetition. Numerous responses indicate that candidates do not know how to write a *while* loop or how to specify the conditions of a *while* loop or an *if* statement. Candidates have difficulty in integrating the different programming structures into a working algorithm. They also need to appreciate the difference between a *for* loop and a *while* loop.

SECTION C – Program Development

Question 11

This question tested candidates' knowledge of graphical user interface objects and their ability to choose appropriate user interface objects to solve a given problem. The question was answered well by most candidates, many of them getting full marks. While there were a few candidates who designed the dialog box with inappropriate graphical user interface objects, the majority designed the dialog box consisting of the following objects:

- Fonts that were used to create labels and the window title
- Text field to enter the first number, labelled with a label object
- Text field to enter the second number, labelled with a label object
- Text field to display the larger of the two numbers, labelled with a label object
- A Button that causes the program to display the larger of the two numbers
- A Window to contain all the other objects

Question 12

This question was designed to test candidates' knowledge and understanding of the queue abstract data type (ADT). The queue ADT removes elements from the top of the queue using a *dequeue()* operation while elements are added to the end of the queue using an *enqueue(Object)* operation. The question was generally answered badly by most candidates with Part (a) and Part (b) being answered better than Part (c) and Part (d). Since new elements are added to the end of the queue, clearly 10 is the answer for Part (a). Two ADT operations for a queue are *enqueue (Object)* and *dequeue()*. An algorithm to remove the elements of *Q1* and place them in *Q2* is given below:

```
while not Q1.isEmpty() do
     set element to Q1.dequeue()
     Q2.enqueue(element)
end while
```

In the future candidates are encouraged to gain practice in writing algorithms that employ ADTs such as queues, stacks, and linked lists.

Question 13

This question was designed to test candidates' ability to write the source code for a class in an object-oriented programming language. It required candidates to declare three instance variables of different data types and to code two methods with simple functionality. Candidates were also expected to write a constructor for the class given. The majority of candidates performed very badly in this question indicating that most of them are not getting adequate exposure to practical object-oriented programming. Nevertheless, a few candidates performed quite well and obtained high marks in this question. A number of candidates mixed and matched language features from different programming languages. In the future, candidates should indicate which language they are using and use only the features of this language in their responses.

Question 14

This question tested candidates' knowledge of tools that can be used to simplify the process of creating computer programs. The tracer was given as an example. Other tools include editor, stepper, linker, and debugger. Most candidates answered this question reasonably well. However, some candidates identified software engineering tools although from the example of the tracer, it was clear that programming tools were required.

Question 15

This question was designed to test candidates' knowledge and understanding of the stack abstract data type (ADT). Candidates were required to find the sum of every other element in a stack using only stack ADT operations. Like Question 12 involving the queue ADT, most candidates answered this question very poorly. However, there were a few rare responses that specified a correct algorithm such as the following:

```
set sum to 0
while not S.isEmpty() do
    set element to S.pop()
    add element to sum
    if not S.isEmpty() then
        S.pop()
    end if
end while
```

A few candidates attempted to define the stack ADT operations in their responses to this question. However, the question required candidates to *use* the ADT operations to solve the problem given. It was also clear that many candidates had great difficulty in writing algorithms involving the fundamental programming constructs of sequence, selection, and iteration. If candidates do not master these concepts, it will be impossible for them to perform well in the questions involving the writing of algorithms or programming code.

PAPER 02

Section A – Software and Software Development

Question 1

This question was designed to test candidates' in-depth knowledge and understanding of two software process models, evolutionary development and the reuse-oriented approach. The question also tested candidates' ability to draw a level-0 data flow diagram from a given narrative. The question was answered fairly satisfactorily by most candidates; however, there is considerable room for improvement in answering this type of question. For example, in the data flow diagram, symbols were often mixed up, indicating that candidates did not fully understand the notation used in data flow diagrams or they lacked experience in applying the notation. The numbers of the processes were often omitted and the data flows were generally incorrect especially with respect to their direction or description of data.

In Parts (b) and (c), candidates demonstrated a good understanding of the evolutionary approach and the reuse-oriented approach for developing software. However, the weaknesses of the evolutionary approach were not properly discussed in Part (b). Similarly, in Part (c), the discussion of the reuse-oriented approach was often not as thorough as was expected. Candidates are reminded that in answering the questions of Paper 2, they are expected to give detailed responses to questions having keywords such as "discuss" or "describe".

Question 2

This question was designed to test candidates' in-depth knowledge and understanding of tools that are typically used in the analysis phase of software development. In particular, it focused on entity-relationship modeling and the use of data dictionaries to formally define data in an organization. Part (a)(i) of the question was fairly well answered by most candidates who attempted this question. Few candidates described the attributes of entities as being a component of the model and some incorrectly stated that the cardinality of relationships is a main component of the model. Part (a)(ii) was poorly done by most candidates. Sources of information to draw an entity-relationship model include computer displays, reports, data flow diagrams, and business documents such as forms. The performance in Part (a)(iii) was fair. Most candidates were able to identify at least three entities from the scenario given. Relationships between entities were generally drawn correctly with a straight line connecting the entities; however, in some cases, no name was given to the relationship.

Part (b) was generally well done by most candidates. They correctly explained that a data dictionary is like a database or repository of data that is important to the organization (that is, a database of metadata). Many candidates obtained full marks for correctly describing the contents of an entry in the data dictionary.

Question 3

This question was designed to test candidates' knowledge and understanding of programming languages and the steps of the object-oriented process. It also tested candidates' ability to write and use recursive functions. Part (a) was badly done by most candidates. Most of them were unable to write the function properly with the necessary arguments or call the function with the right parameters and capture the value returned. They were also unable to return the value of -1 if the search was unsuccessful. Candidates need to improve their skills in defining and calling functions with the appropriate parameters. They also need to understand how to design a recursive function.

Parts (b) and (c) were generally well done. However, in Part (b), there were candidates who described the stages of the general software process instead of the specific stages of the object-oriented process. Also, even though good answers were given for the maintainability aspect of Part (c), many candidates failed to mention that the ripple effect of errors is minimized in a maintainable program and that new changes would not usually cause existing code to malfunction.

Question 4

This question was designed to test candidates' knowledge of various aspects of programming languages. It also tested candidates' ability to write procedural algorithms involving repetition statements (such as *for* loops) and simple data structures such as arrays. The majority of candidates who answered this question omitted Parts (a) and (b), indicating their inability to write algorithms to solve a given problem. Those who attempted Parts (a) and (b) incorrectly specified the *for* loop. They were also unable to iterate through the array (by using the loop index variable) to move or swap the elements. Part (c) was bookwork and was fairly well done by most candidates. However, in Part (d), a large number of candidates described the software development life cycle instead of describing the hierarchical decomposition of programs into modules and the use of the fundamental control structures of sequence, selection, and iteration. Finally, in Part (e), it was expected that candidates would describe the use of the compiler or interpreter to translate a program and the subsequent execution of the compiled or interpreted code on the computer.

Question 5

This question required candidates to identify the graphical user interface (GUI) objects that would be used in a simple program as well as to identify the GUI events that the program needs to anticipate and explain how these events would be handled when they occur. In Part (a), candidates were able to identify the GUI objects that would be used, but many of them had difficulty explaining the purpose of the different GUI objects. In Part (b), several candidates were unable to draw an appropriate GUI with a pleasing layout that one would expect from a modern GUI application. A few candidates even drew several GUI objects without placing them on a window. In Part (c), many candidates were unable to describe two events that the program should anticipate. One event that the program should anticipate is the user clicking on a "Calculate" (or similar) button when he/she is ready to perform the interest calculation. Another event is the user pressing the "Return" or "Enter" key in one of the text fields.

In general, Parts (d) and (e) were badly done and many candidates did not answer Part (f) at all. The code written in these Parts was of a very poor quality, highlighting candidates' inability to write code for fairly straightforward problems. In Part (d), most candidates were unable to write the code to populate *L* with the required numbers. In Part (e), many candidates did not retrieve the data entered by the user in the text fields before doing the calculation of simple interest. For example, in Java, a *getText()* method would be called to obtain the data entered in the different text fields and this data would then be converted to the required type such as *int* or *double* before calculating the simple interest. In Part (f), after calculating the monthly installment, a *setText()* method (in Java) would be used to put the value calculated in the relevant text field.

Question 6

This question tested candidates' ability to write code for an object-oriented program. Candidates were required to write the code for a *Product* class with given instance variables and methods. They were also required to write code to create instances of the *Product* class and to store these instances in an appropriate data structure such as an *ArrayList* or *Vector* (in Java). Many candidates were unable to correctly write the code for the *Product* class. Some candidates did not know how to write an accessor, a method that simply returns the value of an instance variable. Many candidates were unable to correctly reference the *Product* objects created in Part (b) and put them into the data structure specified. An iterator would then be used to access the *Product* objects in the data structure. By invoking the *reorder()* method on each *Product*, it can be known whether a certain product has to be reordered. In Part (c), candidates were expected to describe how the technique of inheritance could be used to create a subclass of the *Product* class. This subclass would automatically inherit the instance variables and methods of *Product*. Inheritance allows additional instance variables to be declared and it also allows the *reorder()* method to be implemented differently in the subclass.

PAPER 03

INTERNAL ASSESSMENT

GENERAL COMMENTS

The quality of the Internal Assessments for Unit I in the 2006 examinations was much better than last year. In Unit 2, most candidates made at least a fair attempt at the Internal Assessment. A few schools used the old syllabus and this presented some problems.

Generally, most candidates chose appropriate topics for the Internal Assessment. The topics chosen were relevant to the level of the candidates and the specific objectives of the respective syllabuses.

One centre submitted samples that were based on the same scenario for a batch of candidates. Teachers should note that each candidate should generally use a different scenario from others for their projects.

DETAILED COMMENTS

Unit 1

Many candidates did not identify a problem in the 'Problem Identification/Situational Analysis' section but rather in the 'Description of the current system and procedures' section. Candidates are asked to desist from including information meant for one section in another section. Teachers should carefully apply the marking scheme when marking candidates' internal assessment reports. They were many instances where full marks were awarded to sections that were not present in the report, for example, some candidates omitted the 'Problem Identification/Situational Analysis' section and still obtained full marks in that section.

<u>Description of the purpose of the study</u>

In this area, many candidates did not address the purpose of the study properly especially with regard to the guidelines given in the syllabus, for. example, response times, volume of data to be stored and processed, and tolerance levels. Descriptions of tolerance levels were particularly weak.

<u>Solution Process</u>

Many candidates used data collection such as interviews and questionnaires but most did not analyze the responses from the questionnaires.

Analysis of Solution

- Hardware Specification: Many candidates stated instead of describing the systems specifications, for example, instead of describing hardware components, some candidates simply provided a listing and many teachers awarded full marks.

- Software Specification: Many candidates simply listed examples of various types of software and provided no descriptions. Network layout diagrams were generally missing.
- Evaluation of Solution: Some candidates attempted to justify their choice of networking components/ hardware/software absolutely rather than relative to other possibilities. In the case of networking, a good response was that a bus network is generally cheaper to set up than a star. Justification of compatibility of network, hardware and software was virtually non-existent.
- Presentation: Diagrams were fairly ok.

Organizational Impact

- Enhancements: Much research was not done on other software that existed on the market. Too many candidates used Microsoft Access® for databases rather than software with a "proven" track-record for the specific functions required.
- Negative Impact: Fairly well-done
- Identification of strategies for addressing issues: While some candidates were able to address this area fairly well, the 'public relations' and 'advertising' sections were either misunderstood or not attempted at all.
- Justification: Some candidates seemed to have been oblivious to the fact that they needed to justify strategies used in the section above.
- Presentation: Many candidates did either an inadequate executive summary or none at all. Quite a few failed to produce a proper contents page and some manually numbered the pages with a pen.
- Communication of information: Candidates need to proof-read their work since there were grammatical errors in many reports.

RECOMMENDATION

Internal Assessment reports should be typed in black ink using a suitable font with double line spacing.

Unit 2

Most parts of the project had at least fair attempts. The 'System Models' section was not properly done. It should be noted that the 'System Models' section of the internal assessment is an important area in Computer Science. This section shows candidates' ability to apply rules to a given situation to create models for analysis. In many cases the rules for drawing entity-relationship diagrams and data flow diagrams were violated. Many context diagrams included data stores, which is a major error. Some candidates mixed up level 1 and context data flow diagrams. It is also clear that some candidates have not been using the recommended texts from the syllabus. Projects were generally well-presented except for a few cases where the overall project presentation could be improved by the use of double spacing and the use of black (not colored) ink for text. In most cases, candidates were unable to give appropriate responses for the non-functional requirements of the project. This information can be found in books such as the Software Engineering text by I. Sommerville.

RECOMMENDATION

Teachers need to provide more guidance for candidates as they work through the SBA. For example, teachers can teach the material for the various parts of the SBA and then see how the candidates apply

CARIBBEAN EXAMINATIONS COUNCIL

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION MAY/JUNE 2008

COMPUTER SCIENCE (REGION EXCLUDING TRINIDAD AND TOBAGO)

COMPUTER SCIENCE

CARIBBEAN ADVANCED PROFICIENCY EXAMINATIONS

MAY/JUNE 2008

INTRODUCTION

This is the eighth year of open examinations for Unit 1 and the fifth year for Unit 2. There were three examination papers in both units, namely, Paper 01, Paper 02, and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the Internal Assessment, was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of multiple choice questions that were designed to test candidates' breadth of coverage of the syllabus. On the other hand, Paper 02 consisted of essay-type questions that were designed to test their depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade remained 50 per cent, 30 per cent, and 20 per cent, respectively.

GENERAL COMMENTS

In general, performance on both units of the syllabus has continued to improve. In Unit 1, 80 per cent of the candidates obtained grades I to III. In Unit 2, 70 per cent of the candidates obtained grades I to III. However, the performance on the School Based Assessment should have been better. Better performances on the SBA would lead to better overall performances on both units as well as better performances on the theory papers. Candidates need to maximize the opportunity to get higher marks on the SBA.

Even though the performance in Unit 2 has improved, there is still concern about the level of programming ability being demonstrated in Sections B and C of both written papers. Candidates continue to find it extremely difficult to write even simple programs, in both the imperative and object-oriented programming languages. Teachers are encouraged to have several programming labs and exercises done with the candidates.

As recommended last year, candidates are encouraged, as part of their examination technique, to read questions carefully before answering, and to respond with sufficient detail that is commensurate with the marks indicated in the question.

DETAILED COMMENTS

UNIT 1

PAPER 01

SECTION A – Components of Computer Systems

<u>Question 1</u>

This question tested candidates' knowledge of extranets, intranets, ROM, RAM, deadlocks and system software.

For Part (a) most candidates obtained only half of the allotted marks. They understood the difference between internet and extranet but they did not adequately differentiate between the two.

For Part (b) while the candidates were able to identify the importance of ROM and RAM, they confused both and as a result they mostly obtained half the mark. They did not adequately identify the software associated with each type.

In Part (c) most candidates obtained more than half the mark. Most answers showed that the student applied their knowledge appropriately and gave several responses that demonstrated a higher level of thinking.

Part (d) was generally well done but in some cases they were suggesting categories of software instead of giving precise examples.

Part (e) was poorly done. Candidates were unable to explain exactly what constituted a deadlock in computer terms. A possible explanation could have been that a deadlock refers to specific conditions when two or more processes are waiting for each other to release a resource. Candidates did not provide adequate examples.

Part (c) (ii) was poorly done. Candidates had a tendency to give definitions of an interrupt but were unable to explain how an interrupt is handled whenever it occurs in the operating system of a computer. A typical solution involves stating that the operating system saves the state of the current process, in execution, stores information related to the new process, runs the process and then restores the original register values in order to return to the execution of the previous process.

Part (c) was attempted by the majority of but many only obtained two marks out of four. A number of candidates provided file management utilities as well, and hence did not obtain full marks. A good response would mention fusing bad disk cluster errors and defragmenting.

<u>Question 2</u>

This question tested candidates' knowledge of

- The role of the OSI model during the transmission of a text file from one computer to the next
- HTTP, FTP, client/server and peer-to-peer as it relates to the internet

Part (a) was poorly done. Candidates were able to produce a diagram of the OSI model but showed little understanding of the workings of the model. Candidates were unable to explain the functions of each layer.

For Part (b), candidates were required to differentiate between 'HTTP' and 'FTP'. Most candidates wrote what they stand for instead of giving their functions.

A few candidates understood the concept of the approaches that can be used to share files over the internet for both client/server and Peer-to-Peer. Most candidates seemed not to understand that the client/server and Peer-to-Peer are two different network architectures and that the terms are not interchangeable.

Few candidates were able to explain that distributed network configurations are fault tolerant and that they have extensive processing power. Most confused the distributed network configuration with the centralised configuration.

SECTION B - Application of Computers

<u>Question 3</u>

Part (a) of this question presented a scenario and candidates had to discuss economic and social implications of using vending machines in a business. Part (b) of this question also presented a scenario and candidates had to suggest advantages of using computer applications to store customer records.

For Part (a) (i) and (ii). These parts were generally fairly well done. However, a number of candidates seemed not to know the difference between economic implications and social implications. Most candidates did not "Discuss" the implication.

For Part (b)(i) Most candidates were able to respond adequately to this part.

For Part (b)(ii) The majority of candidates were able to identify advantages of computer applications but did not go on to discuss these advantages and so were unable to gain full marks. The two main advantages identified were quick retrieval of data and easy boot up of data.

Responses to part (c) varied due to the fact that it was subjective. As a result, most candidates comfortably gained marks.

Almost all responses to part (d) identified two ways to control unauthorised disclosure but most were lacking when it came to the discussion.

The majority of candidates did not gain marks for part d (ii) because they failed to identify an additional strategy other than password or what was identified in d (i). A possible strategy is the use of remote biometric scan e.g. retina scan with remote validation.

<u>Question 4</u>

For Part (a) Candidates were required to explain what is meant by real-time updating of files in a computer application and outline one situation where this approach is necessary. This was well done by most of the candidates. However:

- Some candidates did explain what real-time updating of files is but failed to give any example where this approach to updating is necessary.

- Some candidates' definitions of real-time file update were incorrect. However, the examples given were correct and therefore candidates were awarded some of the marks.

- There were few candidates who did not understand the concept by the reflection of their answers which were totally inappropriate or incorrect.

For Part (b) candidates were required to explain the meaning of data loss, data corruption and unauthorised access as it relates to problems with the computer system at the retail store.

Most candidates did not explain what unauthorised access is, however, some candidates had problems differentiating data loss from data corruption and simply gave answers such as "Data loss as it suggests is loss of data in a computer system."

For Part (b) (ii) in each of the problems, candidates were required to suggest one way in which the computer consultant may have determined the existence of that problem. For data corruption if it could have been determined if the contents of a file were changed.

Some candidates redefined data loss, data corruption and unauthorised access, instead of showing how the consultant might have determined the existence of that problem.

For Part (c) (i) Candidates were required to describe two strategies that can be used to back-up critical data in a government organization that is located in a country in a region prone to hurricanes.

Most candidates just stated the two strategies and did not describe them and therefore could only gain a fraction of the marks awarded.

In Part (c) As a problem solving exercise, candidates were required to determine the steps required to be taken after a hurricane in order to restore the computer systems and data. A vast number of candidates made no reference to the problem solving steps but demonstrated their understanding of the step by the solution they provided.

SETTING - Computer based Problem Solving

<u>Question 5</u>

The majority of candidates scored between 20 and 30 marks. Part (a) was generally well done. Parts (b) and (c) sought to test candidates' appreciation of the qualities and characteristics of information and information sources; these parts were reasonably well done. For part (a) possible responses could have included:

(i) It is much more convenient for readers to access articles from their office or home computers instead of having to wait on the mail or go to the library.

Readers have access to a wider range of articles at a very reasonable price compares to paper subscriptions or library subscriptions which are limited by cost.

(ii) Managing subscription and distributing articles is more convenient with digital technology. Articles are placed in the library and readers only have to download articles from the library.

Publishers reduce time and cost by not having the traditional long delays of publishing. The articles are more attractive to read since they are less out-of-date by the time they are published online.

<u>Question 6</u>

This question tested candidates' depth of knowledge of database and spreadsheet applications software, architecture and usages.

Candidates would have had to read widely and pay attention to formulae to answer the question. Many candidates seemed to have relied on the auto sum function and were not current with the formulae.

UNIT 2

PAPER 02

SECTION A – Software and Software Development

Question 1

Part (a) of this question tested candidates' ability to draw a data flow diagram (DFD) and their understanding of the different features of the rules governing the creation of DFD's. Part (b) tested candidates' knowledge of the properties of well-engineered software. Part(c) tested candidates' knowledge of the advantages and disadvantages of evolutionary development over the waterfall approach.

A significant number of candidates attempted part (a) of the question but many candidates were not able to differentiate between a context diagram and a level-o diagram.

A context diagram is an overview of flow of information to and from the external entities to the system. The context diagram contains no data stores and only one process which is the system.

The level-o diagram contains more details and is a more detailed view of the context diagram. Here the flow of information between the external entities, data stores and processes are shown.

Candidates also had problems correctly labelling data flows where data flows were labeled using verbs rather than nouns and adjectives.

Many candidates seemed to have been exposed to only one data flow model and therefore incorrectly identified the error in the diagrams as being an incorrect symbol rather than the presence of a process in Figure and the absence of input to the process in Figure 1.

In Parts (b) most candidates did a very good job at discussing the two properties of well-organised software.

For Part (c) answers given by candidates, for one advantage and one disadvantage of evolutionary development over the waterfall approach were satisfactorily answered. However, some candidates ignored the word discuss and therefore were not able to gain full marks as they did not explain their points.

Question 2

In Part (a)(i), most candidates were unable to identify the two sources of information for drawing an ERD.

In Part (a)(ii), most candidates were able to construct a proper ERD. In some instances they were unable to establish proper cardinality and relationships. This was rare however. Few candidates drew a DFD instead of an ERD but were still able to identify the entities.

Part (b) which tested candidates knowledge of graphical over interface, was fairly well done by most candidates. However, there were instances where candidates tended not to answer the question within the context of the given scenario. Instead, they attempted to discuss general features of a user interface.

In Part (c), most candidates obtained 2-3 marks out of 6. This was due to the fact that they often gave three reasons why users and management needed to be involved in the development of a software product, which amounted to the same point. They mainly discussed the point that "managers and users are able to specify their needs", in different ways.

SECTION B – Programming Languages

Question 3

Part (a) tested candidates' ability to write on algorithm to reverse the order of the content of an array. Many candidates were not able to gain full marks, due to the size of the dataset. Candidates were able to give a specific solution rather than a general solution. They were not able to swap the elements from location 5 to 9 correctly.

Part (b) was also poorly answered. Candidates were unable to manipulate the indices of the given array. This followed the candidates in arranging even numbers to the front of the array and adding numbers to the back of the array. They were unable to write a proper statement determining whether an integer is odd or even.

Part (c) which tested candidates' knowledge of First and Second generation programming languages was generally well answered.

Part (d) was also well answered, candidates were able to identify control constructs in structured programming but were unable to give proper examples.

Question 4

This question tested the candidates' ability to:

(a) manipulate predefined functions to produce desired output.

(b) manipulate a list of values to produce the largest values using recursion.

(c) apply programming languages to particular situations.

Part (a) was generally well done. Most candidates did not assign values to the parameters and output string.

Part (b) posed a great deal of difficulty for most candidates. Candidates focused on programming languages rather than what the applications created would be used for (cell phone and desktop computer).

In Part (c), most candidates did not attempt this part of the question. Responses reviewed showed limited programming skills and lack of knowledge of recursive functions.

Part (d) was generally well done by the candidates, but most were not able to fully explain how objects communicate by working methods on other objects to which they have no reference. A possible solution for part d (ii) is:

Objects communicate in an object oriented program by message passing. If an object, *A*, of a certain class needs to communicate with an object, *B*, of another class, it first obtains a reference to *B*. *A* then communicates with *B* by sending messages.

SECTION C – Program Development

Question 5

Part (a) of the question tested candidates' knowledge of graphical user interface objects. Candidates for the most part were able to identify correctly the different objects that were used, but many of them had difficulty explaining the purpose of the different GUI objects.

In Part (b) and (c), many candidates demonstrated that they were not aware of what an event is in Event Drives Programming. A satisfactory answer would include description of clicking the calculate button to trigger tax calculation, clicking the exit button to trigger exiting of the window or pressing the enter key in a text field. Simply typing words in a text field is not considered an event as the API would take care of this.

In part (d) (i) many candidates were able to distinguish between unit testing and system testing.

For Part d (ii) candidates were unable to outline the tests that could be performed to **unit test** the Account class. One of the test that candidates could have given is:

Create a new account with a certain balance. Invoke the **get balance** method right afterwards and ensure that the amount returned is the same as that used to create the account.

Question 6

Part (a) tested candidates' ability to write source code for a class in object oriented programming language, it also required candidates to declare instance variables of different data types and code two methods with simple functionality. Many candidates were unable to correctly write the code for *Employee* class. Some candidates did not know how to write an accessor, a method that simply returns the value of an instance variable.

In Part (b) candidates were expected to write a fragment of a code to perform a given test. The majority of candidates performed poorly in this question indicating that most of them are not getting adequate exposure to practical object – oriented programming.

Part (c) required candidates to explain how the *Manger* class could be derived from the *Employee* class. This was poorly done by candidates. Candidates could have stated that the easiest way to create the *Manager* class is to reuse the functionality of the *Employee* class by using inheritance to automatically have the instance variables and methods of *Employee*. In the subclass, additional instance variables such as secretary are simply declared as instance variables in the subclass. Additional functionality is implemented as new methods in the subclass, perhaps by overriding methods such as raise *Salary*.

PAPER 03

INTERNAL ASSESSMENT

GENERAL COMMENTS

The performance on the Internal Assessment was generally good. The projects submitted by some candidates were deficient in various aspects and were still being awarded high marks by teacher.

Teachers need to become more closely involved in the supervision of the projects.

Generally, most candidates chose appropriate topics for the Internal Assessment. The topics chosen were relevant to the level of the candidates' ability and the specific objectives of the syllabus. The treatment of the topics by candidates was adequate. A small percentage was comprehensive though some tended to be superficial. The reports were also generally well presented.

DETAILED COMMENTS

INTERNAL ASSESMENT

Candidates should make more of an effort to follow the layout given in the syllabus. This allows the candidates to clearly identify which parts of the SBA are being responded to. Teachers need to pay particular attention to the symbols that are used in the respective diagrams.

Teachers also need to ensure that the candidate's code is printed and included in the SBA response.

The use of an object oriented language in a non object-oriented fashion should be avoided. Marks are awarded for the use of appropriate classes and class method, thus the use of an object-oriented style of programming.

Few candidates are confusing and mixing functional and non functional requirements.

Any soft copy that is submitted should be given in CD format. Avoid using 3 ½ floppy disks.

There were a few cases where candidates used the old Unit 2 syllabus to prepare their internal assessment. As a result, irrelevant information was submitted.

In some cases, the functionality of the program written was poorly described and there were no screen shots of the working system producing stated functionality displayed in the reports. The scope of the programs was too large, thus the stated functionality seldom matched the actual program produced. Teachers should encourage their candidates to produce programs that focus on simpler more specific problems from the start, allowing candidates to realistically produce what is proposed in the early documentation.

RECOMMENDATIONS

(i) A workshop review of the 2008 examination should be done in each school to assess and detail the concerns and challenges of candidates with particular topics.

(ii) Schools need to carefully compare the new information technology syllabus with the computer science syllabus and decide which of these subjects to offer based on the resources they have available and the interests of students. Schools are advised not to offer computer science if adequate teaching resources are not available.

CARIBBEAN EXAMINATIONS COUNCIL

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION MAY/JUNE 2009

COMPUTER SCIENCE

COMPUTER SCIENCE

CARIBBEAN ADVANCED PROFICIENCY EXAMINATION

MAY/JUNE 2009

INTRODUCTION

The revised syllabus was followed this year for the first time. There were three examination papers in both Units 1 and 2, namely, Paper 01, Paper 02, and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the Internal Assessment, was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of multiple-choice questions that were designed to test candidates' breadth of coverage of the syllabus. Paper 02 consisted of essay-type questions that were designed to test their depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade are 30 per cent, 50 per cent, and 20 per cent, respectively.

GENERAL COMMENTS

In Unit 1, eighty-three per cent of the candidates obtained Grades I to IV. In Unit 2, sixty-four per cent of the candidates obtained Grades I to IV. However, the performance on the School Based Assessment should have been better. Better performances on the SBA should lead to better overall performances on both units as well as enhanced performances on the theory papers. Candidates need to maximize the opportunity to get higher marks on the SBA.

Overall, there is concern about the level of programming ability being demonstrated in both Units. Candidates continue to find it extremely difficult to write even simple programs. Teachers are encouraged to have several programming labs and exercises done with the candidates.

Candidates are also encouraged, as part of their examination technique, to read questions carefully before answering, and to respond with sufficient detail that is commensurate with the marks indicated for the question.

DETAILED COMMENTS

UNIT 1

Fundamentals of Computer Science

Paper 01

The performance on the forty-five multiple choice items on this paper produced a mean of 53 out of 90 with scores ranging between zero to eighty-six.

Paper 02

Section A – Computer Architecture and Organisation

Question 1

This question tested candidates' knowledge of logic gates, truth tables, multiplexers, and internal representation.

Part (a) was fairly well done. However, some candidates failed to include inputs and outputs for the gates. There was also some confusion between AND as well as OR gates. There were some incomplete truth tables and misunderstanding of the NOT gate as well.

Part (b) was poorly done with only a few candidates being able to show the necessary combination of gates. Some gave truth tables instead of the circuit diagram required.

Part (c) was not attempted by many candidates. Teachers need to spend more time on multiplexer design.

In Part (d) (i), some candidates showed a lack of understanding of the binary place values. Part (d) (ii) was fairly well done but some had problems completing the addition. Many did not respond to whether the result could actually be stored. Signed magnitude given in Part (d) (iii) was hardly understood. More practice is needed with these types of numerical questions.

Part (d) (iv) was poorly done and most candidates were unable to get the one's complement so that the subsequent two's complement representation was also incorrect.

Question 2

This question tested candidates' knowledge of word size, cache memory, clock speed, ROM and RAM.

Part (a) tested candidates' knowledge of word size, cache memory and clock speed. Most candidates were able to explain cache memory, but word size and clock speed were not generally well explained. These are concepts that should be reviewed. Additional definitions can be obtained via the Internet.

In Part (b), candidates were asked to distinguish between the computer memory concepts ROM and RAM, access speed and access method and volatility and capacity. The ROM and RAM distinction was well answered by most candidates. The access speed and access method distinction as well as the volatility/capacity distinction were not well answered. However, many candidates were able to indicate that capacity involved storage space.

In Part (c) (i), most candidates did not identify the 'instruction set' as a collection of different instructions that the CPU could execute. Many candidates did not know that the 'instruction format' involved the layout of the instruction into fields corresponding to the constituent elements of the instruction.

Part (c) (ii) was fairly well done. The main problem was that some candidates gave examples instead of stating the types of instructions included in the instruction set.

In Part (c) (iii), some candidates neglected to show the OPCODE as part of the 2-address instruction.

For Part (c) (iv), no candidate scored full marks. Many candidates omitted the storage part of the cycle. Few candidates paid attention to the use of direct addressing.

Section B - Problem Solving with Computers

Question 3

This question tested candidates' knowledge of algorithms.

Most candidates attempted Part (a) but some were unable to give proper definitions. Some key words omitted included 'unambiguous', 'precise' and 'logical'.

Part (b) was fairly well done, however some candidates gave examples rather than listing the actual constructs.

Part (c) (i) was based on an algorithm that was given to the candidates. This part was poorly done. Most candidates had difficulty in identifying the concepts 'dry runs' and 'trace tables'.

Part (c) (ii) was fairly well done. Some identified the correct line but were unable to correct the errors.

Part (d) was attempted by many candidates. Many candidates did not identify the correct symbols for flowcharts. There was also some mix-up in the logical sequence of instructions. Some did not show the looping in the flowchart as required for the 'while' construct. Candidates need to have more practice in moving from algorithm to flowcharts and flowcharts to algorithm. One suggestion is to use partial flowcharts of the correct response and ask candidates to fill in the missing parts during class exercises.

Question 4

Part (a) asked candidates to discuss what the 'Identifying and evaluating possible solutions' stage of problem solving would involve for BuyLo. Part (b) required candidates to trace through the execution of a given algorithm and draw the output. Part (c) required candidates to write an algorithm that used repetition to find the sum of all multiples of 7 between 14 (inclusive) and 126 (inclusive).

In Part (a), most candidates listed Buy Lo's actions through all of the stages of the problem solving process, instead of addressing just the stage identified in the question. Some candidates elaborated on what actions (generally) take place during this stage, but did not give the associated examples necessary to be awarded full marks for this question. As a result, candidates responded poorly to this question overall.

In Part (b), candidates generally responded well and most earned more than half the marks.

In Part (c), candidates generally responded well and most of the candidates who attempted this question got at least 3 marks. Different approaches were also taken towards the solution which usually resulted in a correct response. However, some candidates left out "print" statements at the end of their response.

Section C – Programming

Question 5

The question tested candidates' knowledge of the translation process and their ability to write code using if-then-else and loop constructs. Generally, all parts of this question were poorly done.

In Part (a), only a few candidates were able to describe the steps involved in the lexical and semantic analysis stages. In describing the lexical analysis stage, candidates mentioned that tokens are 'derived' but did not adequately explain how these tokens are derived. Many confused syntactic and semantic analysis.

In Part (b), most candidates omitted the signature of the function (there was no indication of the return type or list of arguments) and prompted the user to enter the two values to be compared, rather than pass them as parameters. A significant number of candidates were unaware of the syntax associated with the if-then-else construct.

In Part (c), candidates did not recognize that the use of loops was required. Some candidates were unable to work with files; their attempts to open, read from and write to the file were syntactically incorrect. Very few candidates used the correct data type (float) to store the average.

Question 6

Part (a) asked candidates to describe declarative, imperative and scripting programming paradigms. Part (b) required candidates to distinguish between syntax and semantics. Part (c) tested candidates' ability to implement loops and update records. This question was poorly done by most candidates.

In Part (a), while most candidates were able to describe the imperative paradigm, very few were able to do so for declarative and even less for scripting, with many of them suggesting that scripting is documentation.

In Part (b), many candidates did not distinguish between the two terms. Most candidates described syntax and some semantics but they did not go on to say what makes them different.

In Part (c), many candidates opted not to respond. Most of the candidates who attempted to respond were able to prompt for and read the input data as well as calculate the pay. Beyond that, some candidates attempted to implement loops for reading data for one employee and used *if* statements for updating the required fields. Some also did not update the required fields but simply assigned the new values to a variable. Most candidates were able to print the required information although not always in the required format.

UNIT 2

Further Topics in Computer Science

Paper 01

The performance on the forty-five multiple choice items on this paper produced a mean of 50 out of 90 with scores ranging between zero to eighty-four. Candidates need to familiarize themselves with Network Architecture, Client Servers and protocol, in particular IEEE802.11 a/b and IEEE802.16g.

Paper 02

Section A – Data Structures

Question 1

This question tested ADTs and associated operations. Overall, it was not well answered.

For Part (a) (i), most candidates misinterpreted what was required and most responses focused on discussing the THREE stack operations without paying attention to 'static computer storage'.

Part (a) (ii) was not well answered. The function prototype was missing in many cases. Very often, the top of the stack was not defined.

In Part (b), many candidates were able to give the correct output for the first three iterations of the *for* loop. Some did not know how to draw a proper diagram of a stack.

In Part (c), many candidates provided general responses that did not address the question given. Some explained the LIFO concept instead of indicating how the stack could be used to determine if a string is a palindrome. Determining whether a string is a palindrome is a popular exercise which candidates should be familiar with.

Question 2

This question tested the candidates' knowledge of two search methods: linear search and binary search.

This question was poorly done by most candidates.

In Part (a), some candidates had a basic idea of linear search but had problems expressing the response as a function. There was regularly a lack of knowledge of the C programming language. Many candidates could not show return type, parameters, function prototypes or return a value from the function.

In Part (b), most candidates had no knowledge of the binary search method. Some candidates attempted this question but most of them had the wrong concept of calculating the midpoint. Some used the number of elements divided by two, rather than using 'low' and 'high'.

Section B – Software Engineering

Question 3

Part (a) (i) had fairly good responses with most candidates obtaining maximum marks. Some candidates incorrectly focused on dealing with the phases of the SDLC rather than discussing prototyping and the frequent adjustments made to the prototype by the users in order to develop a final product.

Part (a) (ii) was satisfactorily answered by most candidates with approximately 80 per cent of the candidates being able to give at least one problem of using the evolutionary development approach. The majority of responses dealt with the costs associated with this approach, but other acceptable responses included process visibility, poor structure and special tools and techniques required for rapid prototyping.

In Part (b), the data flow diagram was poorly drawn with many candidates confusing a DFD with an ERD. Many candidates used the incorrect symbol for process, file and entity. Some candidates gave a context diagram rather than a more detailed diagram as was expected by the examiners. Candidates are therefore advised to only give a context diagram where explicitly indicated by the question, otherwise give a detailed diagram inclusive of files, entities and the necessary processes. Many candidates had DFD with data flows not labeled or labeled but direction of flow not shown. Exercises are required to help candidates improve on the drawing of DFDs.

Part (c) was poorly answered by candidates, with many candidates clearly not knowing what a CASE tool is, and its advantages and disadvantages. Many candidates misinterpreted 3 (c) (i) as asking about the advantages of CASE tools rather than the ways in which a CASE tool can be used in software development.

<u>Question 4</u>

Part (a) (i) was a knowledge based question that tested to see how much candidates know on functional requirements. Most of the candidates were able to describe 'how the system behaves'. However, candidates were still unable to identify the services the software should provide and how the system should react to particular inputs.

In Part (a) (ii), few candidates were able to give complete explanations to this part of the question. Most candidates were able to write about fact-finding techniques or feasibility studies but were not able to also mention examination of documents.

Part (b) tested the candidate's ability to draw an ERD based on a particular scenario. Some candidates were able to construct the ERD. What seemed challenging was differentiating between entity and attributes. Some candidates were actually drawing processes which are found in DFD instead of entities. A few candidates however managed to construct a good ERD.

Part (c) dealt with testing based on a scenario. Some candidates ignored the scenario and spoke about general testing procedures. Most were able to score a few marks.

Section C – Operating Systems and Computer Networks

<u>Question 5</u>

Part (a) (i) was fairly well answered as candidates were able to recognise that user accounts can help track user behaviours; use of username and password for authorized users to protect unauthorized user entry into the system, and to identify each user.

Part (a) (ii) was well answered. Candidates who understood the answer to Part (i) were able to distinguish access logs from files that recorded users on the network.

Part (b) (i) was also well answered. Most candidates gained marks in this part.

Part (b) (ii) was poorly answered, most of the candidates had no idea how two out of the four layers transmitted a file.

Part (c) was not well answered. Candidates generally identified IEEE 802.11b as some wireless standard, giving responses such as the use for satellite, and Wi-Fi. Candidates could not demonstrate properly in a topology how data is passed to and from nodes using a wireless medium such as a wireless router and how to depict this properly in a diagram.

Part (d) was also poorly answered. Candidates were able to identify what the acronyms CDMA and TDMA stand for, but were unable to outline the difference between the two access methods. Many gave incorrect responses relating it to mobile phones. Candidates could not link the idea of different ways of accessing the network by the users on transmission channel, frequency range and spectrum.

Question 6

In Part (a), candidates were not clear about the concepts of device drivers although they may actually use device drivers everyday. Most knew that it was some sort of interface between the OS and the device, but few stated that it was a translator/convertor of instructions for the device to understand. Overall - fairly well done.

In Part (b), most candidates knew that a hybrid network was a combination of two or more network topologies, but a few stated only ‘two or more networks’ and did not state topologies. It was well done by most candidates
In Part (c) (i), not many candidates knew the proper purpose of spooling. Most only stated that the document would be put into a buffer/queue, but few stated further points, for example, that the printer would take control of the printing process from there on. Overall - fairly well done.

In Part (c) (ii), most candidates knew that spooling helps the computer to function efficiently by not using up resources, but few stated that this was as a result of data being stored elsewhere in a buffer. Overall - fairly well done.

Part (d) (i), most candidates knew that the menu interface was easier to use due to a list of options, but not many stated that this would make it easier for novice users. Overall - fairly well done.

In Part (d) (ii) most candidates knew that the command interface would function faster due to instructions being typed in directly, but few stated that this would only be beneficial to users who were knowledgeable or versed in using the interface. Overall - fairly well done.

In Part (e), not many candidates were able to correctly identify two interrupts. Most gave graphic card or video card and RAM problems which would affect game installation but not during game play. Most failed to state I/O interrupts and external interrupts, as well as most did not give proper explanations of the interrupts. Overall – not well done.

In Part (f), most candidates were able to state that paging used virtual memory from the hard disk and that some form of swapping between RAM and virtual memory takes place, but few were able to properly describe the process of paging. Not well done by most candidates.

Paper 03 - Internal Assessment

GENERAL COMMENTS

In general, performance of the candidates was good. However, there were still some inconsistencies arising from candidates and teachers not paying attention to the details of the new syllabus.

Teachers need to work closely with candidates on the requirements for the new syllabus, as well as the specific mark scheme given in the syllabus. Teachers must avoid using old mark schemes when a syllabus has been revised.

Internal Assessment

UNIT 1

DETAILED COMMENTS

Problem Definitions

Problem definitions were not well done in some samples. The provision of a brief context and clear ideas of how the problems were manifested in the organization, along with supporting evidence were often not shown. Many candidates provided extensive backgrounds of the organization instead of the problem description. Most candidates did not pay attention to the requirements when they were writing their problem definition.

Narrative and Flowcharts or Pseudo-code

The narrative description of the algorithms was not well interpreted by some candidates. Narratives were supposed to describe what was designed in the flowchart or pseudo-code algorithm. Some algorithms were not properly designed. Candidates seemed to know the structures, but some were unable to use them to produce good flowchart and pseudo code algorithms. Candidates need to pay attention to correctly designing flowchart algorithms. Teacher practice in this area is recommended.

Coding

The majority of the candidates attempted either flowchart or pseudo code algorithms in the internal assessment, however, some of the programs designed did not match the algorithms.

Candidates were asked to write programs using procedural C only. Some candidates chose to use such languages as JAVA, PASCAL, C++ or Visual Basic. This is deviating from the aim of the syllabus. It is imperative that teachers and candidates pay close attention to the syllabus to avoid being penalized during moderation. Candidates are also advised to print code from the compiler directly and not from a word processor.

UNIT 2

GENERAL COMMENTS

In general performance throughout was good. However, there were a few candidates who used the old syllabus instead of the one prescribed by CXC effective for examination May/June 2009.

Candidates were required to write programs using procedural C only. Some candidates chose to use languages such as JAVA, PASCAL, C++ or Visual Basic. This is deviating from the aim of the syllabus. It is imperative that teachers and candidates pay close attention to the syllabus prescribed in order to achieve its goal.

DETAILED COMMENTS

Problem Definitions

Some candidates focused on providing background information and description of the organization instead of concentrating on the requirements (See syllabus Page 31).

Techniques of Analysis

Most candidates were able to name the techniques of data collection and describe how each was performed. However, some failed to give relevant ones.

Data-Flow and E-R Diagrams

Incorrect symbols were often used within these diagrams. As a result, candidates were unable to produce relevant diagrams. In a few cases, the diagrams did not correctly represent solutions to the problems identified.

Functional and Non-Functional Requirements

Most candidates were able to correctly identify functional and non-functional requirements of the system. However, a few candidates used 'Hardware and Software requirements' (for example, Processor speeds or Operating Systems) for this section which was incorrect.

System Structuring

Most candidates produced a system structure but failed to give ones that were relevant to the project they pursued.

User Interface Design

Most interfaces were relevant. Many candidates were able to correctly state the type of interface they would implement but rarely stated the appropriate justification for its use.

Algorithm Design

Some algorithms were not properly designed. Some candidates seemed to know the structures but some were unable to use them to produce good flowchart and pseudo-code algorithms. Candidates need to pay attention to correctly designing flowchart algorithms. Use of symbols should be practised.

Coding

Some used languages other than C. Some samples were submitted without printed programming code and/or screen shots to verify program functionality and a softcopy was submitted in its place. Generally, this part was well done.

Candidates are advised to print code from the compiler directly and not from a word processor.

Testing

Testing usually focused on normal data and tended not to test abnormal or extreme cases.

Recommendations

Each school should review the syllabus in order to assess and detail the concerns and challenges with particular topics. Schools need to network with each other to utilize the resources available to achieve the aims of the syllabus.

CARIBBEAN EXAMINATIONS COUNCIL

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION MAY/JUNE 2010

COMPUTER SCIENCE

GENERAL COMMENTS

This was the second year for which the revised syllabus was followed. There were three examination papers in each of Units 1 and 2, namely, Paper 01, Paper 02 and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the Internal Assessment, was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of multiple-choice questions that were designed to test candidates' breadth of coverage of the syllabus. Paper 02 consisted of essay-type questions that were designed to test candidates' depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade are 30 per cent, 50 per cent, and 20 per cent, respectively.

In Unit 1, 73 per cent of the candidates obtained Grades I–V. In Unit 2, 92 per cent of the candidates obtained Grades I–V. Performance on the School Based Assessment for Unit 1 was roughly the same as in 2009. However, performance on the School Based Assessment for Unit 2 increased by more than 10 per cent compared with 2009.

Performance on questions involving computing programming continues to be poor; however, there have been small improvements in certain types of programming questions. It is clear that many candidates are still not getting the experience they need in writing and testing real computer programs on their own.

Of particular concern is the fact that performance on other types of questions such as those on Computer Architecture and Organization has also been very poor. This indicates a lack of adequate preparation on all aspects of the syllabus.

DETAILED COMMENTS

UNIT 1

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of 51 out of 90 with scores ranging from 16 to 86.

Paper 02 – Essay Questions

Section A – Computer Architecture and Organization

Question 1

This question examined candidates' knowledge of logic gates, truth tables, binary counters, line decoders, and the internal representation of data on a computer.

Overall, the question was poorly done. It is clear that candidates did not master the basics of computer architecture and organization. Some candidates ignored Part (d) which involved the representation of data on a computer. Several others did not attempt the questions on truth tables. This is cause for concern since the representation of data on a computer and truth tables are the most basic and straightforward items in the syllabus.

In Part (a) of the question, most of the candidates correctly identified the logic gates and gave the required truth tables. However, some candidates mixed up the gates or identified the gates as NAND and/or NOR gates.

Part (b) was poorly done by most candidates. Of the candidates who attempted the question, about 98 per cent of the responses were incorrect. It should be noted that the binary counter goes through a prescribed set of states (0–15) upon application of an input pulse such as a clock pulse. After the counter reaches 15, it goes back to 0.

A fair attempt was made at Part (c) (i). However, most candidates did not label the block diagram properly. In Part (c) (ii), candidates demonstrated a clear lack of knowledge of the truth table of a 2-to-4 line decoder. Part (c) (iii) was also poorly done. Most candidates did not attempt this part. Of those who attempted this part, the majority of them gave incorrect responses.

Part (d) (i) was fairly well done. In Part (d) (ii), most candidates were able to perform the calculation but several of them had difficulty in explaining why the result of the calculation could be stored as a 4-bit binary number. Most candidates attempted Part (d) (iii); however, their calculations were incorrect resulting in incorrect responses. Part (d) (iv) was well done indicating that candidates have a very good grasp of two's complement representation.

<u>Question 2</u>

This question tested candidates' knowledge of storage devices, cache memory, instruction set, instruction formats and instruction cycle. It was misinterpreted by most candidates and answers were vague.

In Part (a), most candidates gave qualitative responses such as *moderate*, *low* and *high* when comparing the capacities of ROM, RAM, hard disk and CD-RW. Similar responses were given when comparing the access speeds of the devices. Very few candidates gave a quantitative response in terms of numeric capacities and placing them in order (ascending/descending). Also, instead of comparing the devices based on access *speed*, many candidates compared the devices based on access *method*.

Part (b) was poorly done. Most candidates were unable to explain how cache memory works and to correctly describe one benefit of its use. In Part (c), candidates focused mostly on the activities of the instruction cycle. Most of them were unable to explain what the instruction set of a CPU is, and to describe types of instructions and typical instruction formats.

Section B – Problem Solving with Computers

Question 3

This question tested candidates' knowledge of the basic control structures used in computer programming and the use of these control structures in a given algorithm. The question also examined candidates' knowledge of representing algorithms using graphical techniques such as flow charts and their ability to understand and modify an algorithm based on given specifications. Overall, the question was not done very well and it appeared that many candidates misinterpreted what was required.

In Part (a), some candidates were unable to identify the three basic control structures of sequence, selection and repetition. Some candidates identified them as input, output and processing. This is cause for concern since the three control structures are the building blocks of computer programs which account for roughly half of the Computer Science syllabus. Most of the candidates misunderstood Part (b) and attempted to explain the control structures without making reference to the algorithm given, as instructed in the question.

In Part (c), some candidates interpreted graphical representation to mean drawing images of actual bananas, pineapples etc. It was expected that candidates would draw a suitably labelled flow chart corresponding exactly to the algorithm specified. One problem with the flow charts given was that the flow lines between components were often missing. In some instances, incorrect symbols were used. Many candidates wrote *for j = 1 to 100* in the decision box, which was incorrect; it should have been, is j equal to 100? The loop is achieved by an arrow returning to a previous point in the flow chart.

The responses that were given in Part (d) demonstrated that candidates did not have a very good understanding of tracing through the logic of a fairly straightforward algorithm expressed in pseudocode. As a result, they were not able to make the required modifications to the algorithm resulting in this part being done badly by most candidates.

Question 4

This question tested candidates' ability to trace through a given algorithm to determine its output and to develop an algorithm from a given specification. The question also examined candidates' knowledge of the *Implementation and Review* stage of the problem solving process. Overall, this question was poorly done.

In Part (a), candidates were able to trace through most of the algorithm. However, many of them were unable to produce the correct diagram of an arrow pointing upwards. In general, Part (b) was done fairly well. Most of the candidates who attempted the question scored at least three marks. This is a good sign since it indicates that candidates' problem-solving abilities are improving.

In Part (c), it was clear that many candidates did not read the question thoroughly. They focused on the actual question and ignored the preamble. Hence, they did not answer the question within the context of the video club. As a result, many of the candidates achieved low scores on this part. Also, some candidates described the entire problem-solving process which was not required.

Section C – Programming

<u>Question 5</u>

This question tested candidates' ability to take a specification of a program and to write a C program which achieves the desired functionality. It also examined candidates' knowledge of programming language features and the appropriateness of programming languages for different types of applications.

Part (a) of the question was generally answered poorly by candidates. Most of their responses described types of programming languages and the generations of programming languages. Few candidates gave the correct response that *mobile devices have limited graphical capabilities and processing power so the programming language will offer less graphical features and smaller libraries.*

Part (b) was fairly well done. Most of the candidates were able to correctly differentiate between a *character* variable and a *string* variable. Part (c) was poorly done by most of the candidates. Many of the responses demonstrated poor programming skills and an inability to manipulate files in the C programming language. Some candidates attempted to answer the question using arrays. However, this was unnecessary since the data for each employee could be read and processed in a *while* loop without saving the data on all the employees.

<u>Question 6</u>

This question tested candidates' ability to write functions in the C programming language. It also examined their understanding of what constitutes good programming style.

In Part (a), the majority of candidates were able to correctly identify spacing and indentation as two ways of improving programming style. The use of consistent case (for example, lowercase) when naming variables is another way to improve programming style.

For Part (b), most candidates were able to explain the purpose of a *struct* in C. However, in many of the responses, the idea of a record was not mentioned.

Responses to Part (c) demonstrated a clear lack of knowledge of modular programming in C using functions. Many candidates did not write the function prototype correctly (return type, name of function, followed by parameter list). The code given in their responses often did not have return statements and many of them were not able to call the functions properly; some candidates even specified the types of the variables when making the function call.

UNIT 2

Paper 01 – Multiple Choice

Performance on the 45 multiple choice items on this paper produced a mean of 54 out of 90 with scores ranging from 22 to 84.

Paper 02 – Essay Questions

Section A – Data Structures

Question 1

This question tested candidates' knowledge of abstract data types and their implementation. It also examined candidates' knowledge of manipulating abstract data types to achieve a desired outcome. Most candidates attempted this question. However, the quality of the responses was generally poor.

In Part (a), most candidates had a general idea of what an abstract data type (ADT) is but most of them could not define it as a specification of a set of data and the operations that can be performed on the data.

In Part (b) (i), some candidates confused the singly linked list ADT with other abstract data types such as a queue or a stack. The ADT operation could have been insertFirst (LinkedList, data) or insertLast (LinkedList, data) or something similar. In Part (b) (ii), most candidates responded well. However, a few candidates omitted the pointers to indicate the beginning and end of the list.

In Part (c) (i), most candidates did not realize that the computer storage they were asked to describe was simply the data structures associated with the implementation of a queue (an array and several integer variables to keep track of the beginning and end of the list).

Part (c) (ii) was generally well done. Part (d) was attempted by most of the candidates. However, some of the candidates did not demonstrate the skills required to reverse the elements of the queue using the stack for temporary storage.

In Part (e), most candidates described the attributes of either a stack or a queue. However, few candidates were able to explain that a stack can contain an embedded linked list and thus, operations such as pop() can be implemented by calling the linked list operation, deleteFront().

Question 2

This question tested candidates' knowledge of sorting and searching algorithms. It was generally answered poorly by most candidates. However, there were very good responses from a few candidates.

In Part (a) (i), most candidates did not give a detailed description of how the selection sort algorithm works. It was important to specify how the minimum element from 0 to 9 would first be found and swapped with the element at position 0. It was also important to specify that succeeding iterations would go from 1 to 9, 2 to 9 and so on, until no more elements are out of place, that is, when the 8 element has been put in its correct position.

In Part (a) (ii), most candidates skipped the second pass in which no swap was made. However, this is a valid pass even though the array remains the same as it was at the end of the first pass.

The responses to Part (b) were of a poor quality. The question required candidates to write a program that performs a linear search. Many candidates incorrectly put the 'if not found' test within the *for* loop, instead of the outside. This would cause the message to be printed every time the element in the array is different from *target*.

In Part (c), explanations of binary search were given but this was not required. Candidates were expected to answer that the binary search should be used when the elements of the array are sorted in ascending or descending order.

Section B – Software Engineering

Question 3

This question tested candidates' understanding of various software engineering concepts as well as their ability to draw data flow diagrams corresponding to a given narrative. It was done fairly well by most candidates.

In Part (a), many candidates were able to correctly discuss the attributes of well-engineered software. However, some of them were unable to list the names of the attributes correctly, for example, reliability was often used instead of dependability as one of the attributes. Also, many candidates listed portability as an attribute but were unable to say what it was.

Most of the candidates correctly discussed the need for user involvement in the software development process in Part (b). However, several of them failed to differentiate between the role of users and managers in the development process. In many cases, the manager's role was simply omitted.

In Part (c), many candidates were not able to clearly explain why a software system needed to be upgraded. The expected response was that *the needs of a business change all the time since the business environment is changing and thus the software must be able to adapt to accommodate these changes.*

In Part (d), a few candidates did not know which symbols to use when drawing a data flow diagram. At times, entity-relationship diagram symbols were used. Also, several candidates gave a context diagram instead of the Level-0 diagram requested.

Question 4

This question tested candidates' knowledge of the feasibility study and the requirements specification document in software engineering. It also tested candidates' ability to draw an entity-relationship diagram from a narrative and to specify ways in which a piece of program code can be tested. The question was well-attempted. However, it was poorly answered.

In Part (a), most candidates identified economic and technical feasibility as reasons for undertaking the feasibility study. Few candidates recognized that it may be possible for user needs to be satisfied with existing software and hardware.

Many candidates were able to gain most of their marks in Part (b) of the question. However, several of them omitted the identification of the primary key, for example, underlining the *employee number in* some notations.

Many candidates did not seem to have read Part (c) of the question properly and so did not relate their responses to the parameters of the function *lsearch*. Many candidates simply described general forms of testing and did not apply the principles of testing to the function given. For example, one test is to call *lsearch* with an array of integers which does not contain the given *key*; if *lsearch* is working correctly, it should return -1.

Section C – Operating Systems and Computer Networks

Question 5

This question tested candidates' knowledge of the OSI model for computer communication and various operating system concepts such as page fault and process states. It also tested candidates' knowledge of GPRS. It was generally poorly attempted by most candidates, most of whom obtained under 10 marks, with a significant number obtaining 0–5 marks.

In Part (a), most candidates were able to correctly identify the seven layers of the OSI level. However, a few candidates mixed up the transport and network layers. Some candidates also labelled the layers in the reverse order which resulted in them obtaining fewer marks. Part (a) (ii) was poorly done. Approximately 5 per cent of the candidates were able to correctly describe the purpose of the three layers.

In Part (b), only a few candidates were able to accurately define a page fault and describe how it was handled by the operating system. Most candidates described the paging process and did not answer the question. Many of the responses incorrectly stated that a page fault occurs when there is an error on the page. The correct answer is that *a page fault occurs when a program tries to access a page that is not currently mapped to RAM.*

In Part (c), candidates generally confused GPRS with GPS and consequently answered the question incorrectly. GPRS is a technology used to connect wirelessly to the Internet and can be used by point-of-sale devices or other devices needing wireless connectivity to the Internet. Part (d) was fairly well-attempted. However, many candidates did not clearly describe the steps involved when a process moves from the running state to the ready state.

Question 6

This question tested candidates understanding of various networking and operating system concepts. It was attempted by most candidates. Responses were fair with average scores between 11 and 15 marks. A few candidates got 21 – 25 marks.

Parts (a) to (g) and (k) (iii) were well done by most candidates. Parts (h), (i), (j), k (i) and k (ii) were not answered well by most candidates. They were unable to provide proper explanations which indicate the need for greater depth of knowledge in these areas, for example, client-server versus peer-to-peer configurations.

Based on the responses obtained for this question, it is clear that candidates need to become more familiar with the technical terms used in the syllabus.

GENERAL COMMENTS

Paper 03 - Internal Assessment

In general, performance on the School-Based Assessment was very good. The performance on Unit 1 was roughly the same as in 2009. However, the performance on Unit 2 increased by more than 10 per cent compared with 2009.

DETAILED COMMENTS

UNIT 1

Some candidates wrote their programs using separate program files which were saved and printed individually rather than using separate functions within the same program. There were candidates who did not provide printed evidence for the headings which were included in their submissions, for example, a heading stating 'PSEUDOCODE' often had no printout showing an algorithm in pseudocode.

Some candidates included trace tables which were not required. Teachers need to pay careful attention to the requirements of the syllabus. Other candidates used data flow diagrams instead of narratives for describing algorithms. There were also candidates who utilised one main function and no function decomposition was seen in some samples.

Some candidates submitted a pseudocode which was identical to their programs. A pseudocode should be produced before the program is written and will generally look different from the source code of the program. In some samples, pseudocode algorithms were not written using correct logic, for example, variables controlling iterations were not incremented and *while* statements were not closed with *Endwhile*.

There were candidates who submitted the same pseudocode, program code and test data. Subheadings were often missing in some of the documents submitted and the word limit was exceeded in some instances. Flow charts in many samples made use of system symbols rather than the normal flowchart symbols and some test plans did not use erroneous and extreme data for testing. Screen images (screen shots) to illustrate the program output were not shown in many samples. These are important for verifying the output of the program.

The problem statements in some samples were either very trivial or too detailed. Candidates are reminded that the source code for programs must be printed from the C development environment and that a narrative must be given which is related to the problem solution.

Unit 1 samples often included entity-relationship diagrams and data flow diagrams; however, these are only relevant to Unit 2.

UNIT 2

Some candidates were unable to differentiate between functional and non-functional requirements. Functional requirements are features the system must provide. Non-functional requirements are constraints that apply to the system, for example, it must perform a transaction in less than two seconds.

Attributes of entities were not included in the entity-relationship diagram by most candidates the entity-relationship diagrams and data flow diagrams were often inconsistent with the problem definition.

It should be noted that user technical manuals are not required and that background information and abstract information should be kept to a minimum.

Some problem definitions were often too wordy, and, in some instances, made no sense in the context of the system developed.

Candidates are reminded that all design diagrams must be drawn electronically, for example, system structuring and user interface design. Programs must be written in C code and printed output must be done from the C development environment.

When preparing their reports, candidates should use subheadings which are specified in the syllabus. The ordering of the headings should also be followed.

RECOMMENDATIONS

Candidates are encouraged, as part of their examination technique, to read questions carefully before answering, and to respond with sufficient detail that is commensurate with the marks indicated for each question. Candidates need to answer questions in the context given instead of simply regurgitating notes. Past-paper questions should be carefully studied to understand how syllabus items can be tested.

Teachers need to prepare students adequately for the three modules of each unit. Algorithm Development, Computer Programming, and Data Structures account for three of the six modules of the Computer Science syllabus so teachers need to cover these modules in more detail, scheduling as many practical sessions as possible. This can take the form of

- classroom discussions geared to solving specific problems or developing algorithms to solve given problems
- laboratory exercises involving the writing and testing of computer programs and experimentation with ‘buggy’ programs
- using the Internet to download tutorials on programming and sample code to experiment with.

The poor performance by candidates on the Computer Architecture and Organization module can be handled by adequate coverage of the syllabus items. This module does not involve computer programming. Candidates need more practice with numerical problems such as those involving decimal to binary conversion and vice versa, as well as problems involving one’s and two’s complement.

With respect to the School-Based Assessment, teachers need to provide better feedback to studentss. The School-Based Assessment should be viewed as a process with well-defined milestones. As students submit work for these milestones, teachers need to provide adequate guidance, correcting errors made and indicating clearly what is required for the next milestone.

CARIBBEAN EXAMINATIONS COUNCIL

REPORT ON CANDIDATES' WORK IN THE ADVANCED PROFICIENCY EXAMINATION

MAY/JUNE 2011

COMPUTER SCIENCE

GENERAL COMMENTS

This was the third year in which the revised syllabus was examined. There were three examination papers in each of Units 1 and 2, namely, Paper 01, Paper 02 and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the School-Based Assessment (SBA), was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of 45 multiple-choice questions that were designed to test candidates' breadth of understanding of the syllabus. Paper 02 consisted of six essay-type questions that were designed to test candidates' depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade are 30 per cent, 50 per cent, and 20 per cent, respectively.

In Unit 1, 87 per cent of the candidates obtained Grades I to V, a 12 per cent improvement over 2010. In Unit 2, 87 per cent of the candidates obtained Grades I to V, a five per cent decline compared with 2010. In general, performance on the SBA was very good. In Unit 1, there was a slight decline in the performance on the SBA compared to 2010. The average mark was 68 per cent compared to 70 per cent in 2010. However, there was a significant decline in the performance on the School Based Assessment in Unit 2 compared with 2010. The average mark was 57 per cent compared with 66 percent in 2010. Nevertheless, in both units, the average mark was an improvement on the corresponding 2009 average.

Performance on questions involving computing programming continues to be poor. However, there was a reasonable improvement in scores for most of the programming questions in Unit 1 compared with 2010. Performance on the programming questions in Unit 2 was roughly the same as in 2010. Despite the improvements noted, it is clear that many candidates still need to get more experience in writing and testing real computer programs on their own.

Of particular concern is the fact that performance on other types of questions such as those on computer architecture and organization has also been very poor. This indicates a lack of adequate preparation on all aspects of the syllabus.

DETAILED COMMENTS

UNIT 1 – Fundamentals of Computer Science

Paper 01- Multiple Choice Questions

Performance on the 45 multiple-choice items on this paper produced a mean of 54 out of 90 with scores ranging from 18 to 86.

Paper 02 – Essay Questions

Section A – Computer Architecture and Organization

Question 1

This question examined candidates' knowledge of truth tables, logic gates, circuit diagrams, flip-flops, multiplexors, decoders, and the internal representation of data on a computer. It also required candidates to apply their understanding of a decoder to solve a simple problem.

Overall performance on this question was poor and the majority of candidates obtained less than 10 marks.

In Part (a), only a few candidates were able to correctly convert the truth table to a circuit diagram. Some candidates even used secondary logic gates (NOR, NAND) in their responses which were unnecessary. In Part (b), most candidates performed well and got full marks or at least half of the marks allocated for this question.

In Part (c), many candidates stated that a flip-flop can be used to store one bit of data. However, only a few of them went on to fully explain what is a flip-flop and hence did not obtain full marks. Many candidates obtained full marks for their explanation of the multiplexor.

Part (d) (i) was generally well done. Most candidates gave a correct diagram for the decoder; however, some candidates did not label the diagram and a few attempted to draw a circuit diagram. The performance on Part (d) (ii) was fair. A few candidates drew circuit diagrams instead of using the block diagram of the decoder. Most candidates did not explicitly state that there are two inputs and that the bulbs have to be connected to the outputs. Also, some candidates did not explain how the two inputs were connected to the four outputs (the bulbs).

Part (e) was fairly well done. The majority of candidates were able to convert the decimal value to binary, but a number of them had difficulty finding the two's complement. Part (f) was poorly done. Many candidates simply converted the binary representation to its decimal value. Some of them correctly converted the 3-bit exponent to decimal but were unable to use this value to correctly complete the calculation. A few candidates assumed that the decimal point was to the right (00101.) rather than the left (.00101).

There is cause for concern with candidates' performance on this question, particularly on Parts (e) and (f) since the representation of data on a computer is one of the most basic and straightforward items in the syllabus.

<u>Question 2</u>

This question tested candidates' knowledge of registers, Central Processing Unit components, storage devices, cache memory, instruction formats and addressing modes.

The question was answered fairly well and there was an improvement in responses compared with to those given for a similar question in 2010.

In Part (a), candidates' were generally weak in their understanding of registers found in the CPU. Candidates were expected to mention registers such as the MAR, MBR, PC, AC and IR.

The responses to Part (b) were generally fair. However, many candidates only compared main memory and the two given storage devices in terms of capacity or access speed alone. They thus lost half of the marks for this question.

Part (c) was generally well done since this is a topic well understood by most candidates. The Arithmetic and Logic Unit (ALU) and Control Unit (CU) were popular responses for this question. However, the function of the components stated (e.g. the CU) was not properly stated in many responses.

The responses to Part (d) were generally fair. However, many candidates could not clearly explain how cache memory increases the efficiency of data retrieval (by keeping frequently accessed portions of main memory in the cache).

The responses to Part (e) were generally poor. Candidates need to spend more time understanding the various types of addressing modes specified in the syllabus.

Section B – Problem Solving with Computers

<u>Question 3</u>

This question tested candidates' knowledge of the stages in the problem-solving process and their skills in constructing a flow chart from an algorithm as well as tracing through and identifying errors in an algorithm. The question also examined candidates' ability to understand and modify an algorithm based on given specifications.

Overall, the question was well done and it appeared that many candidates understood what was required. The mean performance on this question was the best of all the questions in Paper 02.

In Part (a), most candidates were able to identify three stages in the problem-solving process. The stages were sometimes incorrectly given as the stages of the systems development life cycle. Candidates were often unable to clearly describe each stage, particularly the analysis stage (which seeks to determine how the problem manifests itself).

Most of the candidates gave good responses for Part (b). However, some of them drew incorrect symbols to represent input/output and processing steps and some were uncertain about how to represent a loop in the flow chart. In a few cases, programming code was included with the symbols. Candidates need to spend more time constructing flow charts and should make sure that they understand the purpose of each symbol used in a flow chart.

In Part (c), candidates were able to correctly trace through and identify the lines of the algorithm containing the errors. However, many were unable to correct the errors. It is recommended that candidates trace through an algorithm after modifying it to ensure its correctness.

Question 4

This question tested candidates' ability to trace through a given algorithm to determine its output and to develop an algorithm from a given specification. The question also examined candidates' knowledge of the properties of a well-designed algorithm.

In Part (a), many candidates were unable to identify three properties of a well-designed algorithm. This was rather unfortunate since it involved straightforward recall of a syllabus item.

In Part (b), many candidates were unable to trace through the while loop correctly and generate the triangular shape of asterisks. In addition, there was some confusion between the use of *print* and *println*. However, the question clearly stated that *println* did the same thing as *print* except that subsequent output started on a new line. Only a few candidates were able to correctly calculate the values for *y* on each repetition of the *while* loop. This indicates a weakness in basic arithmetic skills. Candidates need to spend more time tracing through algorithms since this is a fundamental skill in computer science.

In Part (c), some candidates were unable to write correct algorithms in terms of looping constructs, accumulating the sum, and finding multiples of 11. It was clear from their responses that many candidates were unaware that a *while* loop can be bounded or had a misconception that a *while* loop is always unbounded. A loop is bounded if the number of iterations is known beforehand or can be calculated beforehand. It is sometimes possible to calculate the number of iterations of a *while* loop before it is executed; thus, in these situations, the *while* loop is bounded.

Section C – Programming

Question 5

This question tested candidates' knowledge of the stages of the translation process and of the advantages of using a modular approach in programming. Candidates were required to write a C function which took two parameters, one of which was an integer array. The question also tested candidates' ability to take a specification of a program and write a C program which achieves the desired functionality. The program required candidates to know how to read and write from text files.

Part (a) was answered fairly well by candidates. Most of them were able to list the stages of the translation process.

Part (b) was fairly done. However, a few candidates confused the term *modular* with the *modulus* operation. Also, several candidates were vague or overly general in their responses.

Part (c) was poorly answered. Many candidates gave a program in their response rather than a function and lost marks because of this. It is fairly straightforward to find the sum of the numbers in an integer array and the code for this functionality should have been placed in the body of the function. In addition, candidates were expected to specify the parameter list and return type of the function as well as return the sum of the values after it was calculated inside the function.

Part (d) was also poorly answered. Candidates seemed to lack skills of analysis to understand the functionality that the program was required to provide. For those who were able to write a reasonable program, it was clear that most of them were unable to manipulate text files. Many of the candidates did not adhere to the syntax of the C language, particularly in the use of semicolons, ampersands, format specifiers and appropriate quotation marks. Candidates need to spend more time writing programs from a given specification using the programming language features outlined in the syllabus.

Question 6

This question tested candidates' understanding of good programming style such as the use of white space and proper indentation. It also tested their understanding of the term *debugging*. Candidates were required to write a program to determine what type of triangle is present given three integers representing the sides of a triangle. The key component of the solution consisted of an *if-then-else-if* statement. Finally, candidates were required to trace through a segment of C code to determine its output and to modify the C code to give a different set of output.

In Part (a), the majority of candidates did not obtain full marks since they did not clearly explain white space.

In Part (b), most candidates were able to define the term *debugging* using their own words. The concept of debugging seems to be well understood among candidates.

In Part (c), most candidates had a general concept of indentation being associated with the appearance or presentation of a program. However, the basic idea of maintenance was not present in their responses.

In Part (d), the responses indicated that candidates lacked the ability to properly sequence code in order to achieve the required results. Thus, for example, the check for equilateral triangle should have preceded the check for isosceles triangle. Many candidates were unable to maintain the use of correct syntax throughout their programs. Program code often lacked logical operators in the condition part of the nested *if* statement and several candidates opted to use a series of *if* statements to avoid the use of a nested *if* statement. However, this resulted in incorrect code since one *if* statement would flow into another one. It should also be mentioned that some candidates gave their responses in C++. Candidates need to use the C language in the present Computer Science syllabus.

In Part (e), a number of candidates lacked the ability to trace through the *for* loop so they were unable to generate the output. Some responses ignored the inner *for* loop indicating that these candidates did not understand nested loops. Of those who correctly traced the required output, only a few were able adjust the loop to generate the required output in Part (e) (ii). Again, candidates need to spend more time writing and tracing through programs in order to answer questions like these.

Paper 03 – School-Based Assessment

In general, performance on the Internal Assessment was very good. In Unit 1, there was a slight decline in the performance on the Internal Assessment compared to 2010. The average mark was sixty-eight per cent compared to seventy per cent in 2010. However, there was a significant decline in the performance on the Internal Assessment in Unit 2 compared to 2010. The average mark was fifty-seven per cent compared to sixty-six percent in 2010. Nevertheless, in both units, the average mark was an improvement on the corresponding 2009 average.

Many students `included documentation which was not required for the Unit 1 SBA including structure charts, Gantt charts, data-flow diagrams, questionnaires and entity-relationship diagrams. Many of the projects contained flowchart algorithms; however, system flowchart symbols were incorrectly used instead of programming flowchart symbols.

There were many instances where full marks were awarded for problem descriptions that were either superficial or overly detailed.

Many students did not provide enough detail in the narratives when describing a solution to the problem identified.

Generally, the presentation of pseudocode algorithms showed improvement in logic compared with previous years and were also easy to follow for many of the projects. However, many students provided a pseudocode algorithm that was identical to their C programs. In some projects, the algorithms were written using incorrect logic. For example, variables were often not initialized before being used and repetition statements did not show variables being incremented correctly.

Students often provided programs containing logical steps that were not included in their algorithms.

Some students made use of alternative programming languages such as C++ and Pascal for writing their programs. Schools and students are reminded that C is the language of the Computer Science syllabus.

Programs were often not printed directly from the C development environment and this made it difficult to properly verify the documentation and modularization of the source code. Many students did not make use of comments and did not indent their C code.

Teachers should ensure that programming projects sufficiently cover items in the syllabus such as the use of arrays and record structures as well as the manipulation of external data files.

Trace tables were used in many instances but these were not required for program testing. Program testing by many students often did not show the use of normal, extreme and erroneous data. The test results were often described but not shown by means of screen images. The screen images are required to verify that the program produces the output as described in the report.

Candidate projects should make use of the section headings given in the Criteria for Marking the Internal Assessment in the syllabus. Many projects often did not include any section headings.

The use of GOTO statements should not be encouraged in structured programming.

Storage media such as CDs should not be included with the submission of the samples.

Care should be taken when preparing the SBA document to ensure the following:

- The table of contents is accurate
- Pages are numbered
- Text size, font style and line spacing are legible for reading and moderation.

Teachers should also ensure that each sample has a completed CSCI1-5 form detailing the marks awarded for the assessment criteria.

UNIT 2 – Further Topics in Computer Science

Paper 01

Performance on the 45 multiple-choice items on this paper produced a mean of 58 out of 90 with candidates' scores ranging from 18 to 86.

Paper 02 – Essay Questions

Section A – Data Structures

Question 1

This question tested candidates' knowledge of abstract data types (ADTs) and their implementation. It required candidates to use the Stack ADT to remove the element at the bottom of a stack. The question also examined candidates' knowledge of manipulating ADTs to achieve a desired outcome. In addition, the question tested candidates' understanding of the implementation of a circular queue as well as their understanding of the bubble sort algorithm.

Part (a) required candidates to define the term *abstract data type* (ADT) and to explain how the Stack ADT is implemented. Most candidates had some knowledge of the ADT but were unable to define it properly. In Part (a)(ii), many candidates listed the operations of the Stack ADT but did not explain how the stack would be implemented in code (e.g. by using an array and a *top* pointer and manipulating these data items in the *push* and *pop* operations).

In Part (b), many candidates did not recognize that this was a problem to be solved using the operations of the Stack ADT (e.g., *push*, *pop*, *isEmpty*). Consequently, they wrote unnecessary code to manipulate the internal data of the stack. Thus, they ended up writing code to push and pop the elements from the stack which was not required.

Performance on Part (c) was generally poor. It was clear that many candidates did not understand how a circular queue operates. Some candidates were able to perform the insertions correctly. However, when an element was deleted, many candidates simply shifted the remaining elements one position to the left. In some responses, the size of the array increased or decreased when elements were inserted or deleted, respectively. It should be mentioned that a circular queue operates on an array of fixed size by maintaining pointers to the first and last elements in the queue. Elements are not shifted when inserting or deleting another element.

The quality of the responses for Part (d) was also generally poor. It seems that the majority of candidates did not have a clear understanding of the bubble sort algorithm. Several candidates described the first pass correctly but did not go on to explain what happens after the first pass. They simply concluded their responses by saying that the algorithm is repeated until the entire array is in sorted order. But, getting the array in sorted order is the purpose of the sort algorithm in the first place. Some candidates wrote C code in their responses. However, the question did not require the writing of code.

Question 2

This question tested candidates' knowledge and understanding of linked lists as well as their ability to implement the selection sort algorithm in C. The question also required candidates to demonstrate their understanding of what happens in the linear and binary search algorithms.

In Part (a), most of the linked list diagrams drawn were incorrect. Many candidates gave tables instead of the standard representation of a linked list (a set of nodes pointing to each other with a special pointer to the top of the list and a terminator symbol at the last node). Of those who drew a diagram containing nodes with each one pointing to the next node in the linked list, many did not did not use a terminating symbol at the last node in the list. In terms of the explanation, many candidates described advantages of the storage characteristics of a linked list instead of simply describing its main features.

In Part (b), the implementation of the selection sort algorithm was poorly done. Candidates demonstrated weaknesses in storing values in the array at the beginning of the algorithm, in writing the nested *for* loops properly (especially with regard to the initial and terminal values of the loop variables), in swapping elements in the array; they generally did not know how the selection sort algorithm works.

The responses for Part (c) were generally good. However, many candidates did not clearly state the conditions under which the linear search algorithm would terminate. Several candidates also did not properly describe what happens in each execution of the loop in the binary search algorithm.

This question required a significant number of narrative responses from candidates. The level of English was very poor and it was often difficult to decipher what a candidate was trying to say. Many candidates did not answer the questions directly, but tended to write down everything they knew on the specific topic. The level of programming was also very poor indicating that candidates have not developed their programming skills sufficiently in Unit 1 to master the syllabus items in this module.

Section B – Software Engineering

Question 3

This question tested candidate's understanding of various software engineering concepts as well as their ability to draw a Level-0 dataflow diagram corresponding to a given narrative.

In Part (a), candidates were required to explain the waterfall approach to system development. Candidates were expected to identify the stages of the waterfall approach and explain how one stage fed into the subsequent stage. However, very few candidates associated the waterfall model with the systems development life cycle (SDLC) even though this is one of the most popular approaches for the SDLC. The minority of candidates who made the link did not identify the phases of the waterfall approach.

In Part (b), candidates were required to describe four tasks which must be performed during the design phase of systems development (e.g. architectural design, interface design, data structure design and algorithm design). Only a few of the candidates who attempted this question were able to describe these tasks.

Part (c) required candidates to draw a Level-0 dataflow diagram for a given scenario involving a mail order company. Most candidates attempted this question. However, it seems that many of them were not equipped with "drawing tools". Consequently, many of the diagrams were drawn free hand and ended up being very ugly and unsightly for advanced proficiency candidates. Also, on many occasions, the standard symbols for drawing data flow diagrams were not used.

Question 4

This question tested candidates' knowledge of the techniques used for fact-finding during the analysis phase of systems development as well as the deliverables of the analysis phase. It also tested candidates' understanding of the use of CASE tools in software development as well as their ability to use entity-relationship models for data modelling.

Part (a) (i) was well done overall. Most candidates were able to identify three fact-finding techniques such as interviews, observation and questionnaires. In contrast, Part (a) (ii) was poorly done by most candidates. It was clear that many candidates did not understand the term *deliverables* and as a result only a small percentage were able to accurately state two deliverables.

In Part (b), the responses were generally poor. It was clear that most candidates did not know what a CASE tool is and consequently, their responses for both parts of the question were poor.

Part (c) was fairly well done overall. However, many candidates did not use the correct symbol to denote an attribute of an entity and consequently lost marks for drawing the attributes. Also, candidates had difficulty accurately identifying the many-to-many relationships in the scenario (e.g. a lecturer is hired by many departments and a department hires many lecturers). Many candidates drew lines connecting entities (to represent relationships) but did not label the lines or give the cardinality of the relationships. Several candidates also used non-standard notations to represent cardinality. In the future, candidates are advised to use a standard for their entity-relationship models (such as the Unified Modelling Language or the more dated Chen's notation).

Section C – Operating Systems and Computer Networks

Question 5

This question tested candidates' knowledge and understanding of various networking concepts such as network media, network configuration, networking devices, transmission media and access to data and resources on a network

Some parts of the question were answered fairly well by many candidates. However, the responses to the other parts were often very poor.

Part (a) was not answered well by most candidates. Most of them were unable to correctly identify and explain a difference between an analogue signal and a digital signal. Some rather interesting responses stated that an analogue signal is understood by computers while a digital signal is not.

Most candidates answered Part (b) correctly. However, some candidates gave incorrect answers such as telephone, radio and fax machines. This suggests that these candidates may not be familiar with the concept of transmission media.

In Part (c), some candidates gave good responses by explaining the layout of a client-server network with the aid of a labelled diagram. However, some candidates did not understand what the client-server model is and instead gave answers related to the peer-to-peer model or to network topology such as star or ring.

Part (d) (i) tested candidates' knowledge of network connectivity devices such as bridges, repeaters, routers, and gateways. Some candidates described network *types* instead of network *devices*, while a few described devices that are used in a network such as UPS, server. A few candidates also described a network topology such as star, ring, or bus. This suggested that candidates did not know what network connectivity devices were or did not know the difference between network *topology* and network *devices*.

Part (d) (ii) tested candidates' ability to apply knowledge of transmission media to a specific situation. Responses from candidates revealed that some of them did not know what transmission media is and were thus unable to answer the question correctly. Some candidates even gave responses like cell-phones, server and pager.

Part (d) (iii) tested candidates' knowledge of the different ways that users may be granted access to resources on a network. Many candidates confused the granting of access to resources on the network with preventing access to a network and incorrectly gave responses such as encrypting files and using firewalls. Some candidates also stated that access can be granted by using the Internet, an intranet or an extranet.

Question 6

This question tested candidates' understanding of various networking and operating system concepts. Performance was generally fair.

In Part (a), candidates were often unable to properly or clearly distinguish between a batch processing system and a multi-user system. For example, some candidates stated that in a batch processing system, only one user used the system at a time. The examples provided in Part (a) (ii) and Part (a) (iii) were often not representative enough of the type of system.

The responses for Part (b) were generally good with most candidates giving answers such as passwords and encryption.

The responses for Part (c) were very poor. It is clear that most candidates do not understand the concept of an interrupt and the interrupt processing mechanism.

The responses for Part (d) were also poor. Many candidates did not recognize that the diagram represented a deadlock situation and gave interesting answers such as 'hold ups', 'event interrupt', 'round robin' or even 'parallel processing'.

In Part (e), many candidates were unable to completely describe how the round-robin scheduling algorithm works. It also seemed that weaker candidates had no idea of what was involved in the algorithm.

PAPER 03 – School-Based Assessment

For the Unit 2 SBA, students are expected to choose a problem for which a software solution exists and then develop the software using software engineering techniques. In particular, they are expected to demonstrate the tools and techniques used in the analysis of the software to be developed. They are then expected to design, code and test their software using appropriate techniques.

Performance on the SBA was generally fair. It was observed that the projects submitted by Students were deficient in various aspects and were still being awarded high marks. Teachers also need to become more closely involved in the supervision of the projects.

Generally, most students chose appropriate topics for the SBA. The topics chosen were relevant to the level of the students and the specific objectives of the syllabus. Diagrams were generally done well; however, they were very inconsistent and did not flow (as they should) from one to the other. The treatment of the topics by students was adequate. A small percentage was comprehensive though some tended to be superficial. The reports were also generally well presented.

Students should put more effort in following the layout of the SBA given in the syllabus. This enables students to clearly identify which parts of the SBA are being addressed by which sections of the report.

Teachers should instruct students to use the requirements in the mark scheme as sub-headings within their reports. This ensures that students will have a section corresponding to each requirement.

Teachers need to advise students to restrict the scope of their SBA Reducing the scope of the project or focusing on a single aspect will give students a better chance of completing what is set out in their problem definition.

SBA reports should be well secured and bounded to avoid the mixing up or loss of parts of the reports submitted.

Teachers should not assume that marks are given for things not printed. Only visual/printed elements can be marked so anything not included in printed form within the SBA will not receive marks. Teachers also need to ensure that each candidate's code is *printed* and included in the SBA report. A soft copy of a candidate’s code is not accepted.

Teachers should ensure that students clearly state the problem being tackled in the problem definition. Simply stating that a system is manual or paper-based is insufficient. A manual or paper-based system is not in itself a problem, as stated by many students. The fact that it may be time consuming, error prone, tedious and so on are problems that may exist in a particular manual or paper-based system. Not all manual or paper-based systems suffer from these problems; thus, students should identify the problems clearly in order to be awarded full marks.

In describing the techniques used for analysis of the problem, many students did not say why the techniques were relevant.

It should be noted that a few teachers and students continue to confuse the terms *functional requirements* and *non-functional requirements*. Special attention should be paid to the listing of functional requirements as this was generally done poorly.

Students need to extend their system structuring diagram beyond the elements of the main menu in order to be awarded full marks for this requirement.

Students seldom included justifications for their choice of user interface. Also, some of them were confused as to the difference between menu-driven and command-driven interfaces.

Students in general left out the report design; teachers should ensure that this is included and not assume that marks will be awarded based on screen shots.

Test plans were not done well in general. Students should include an exhaustive set of data to be tested which includes all forms of data entry. Many students only tested the menu selections. Expected results and actual results obtained from testing should be included in tests. Tests should also include normal, abnormal and extreme data.

In most cases, the functionality of the program written was poorly described and there were no screen shots of the working program producing the functionality described in the reports. The scope of the programs was often too large; as a result, the stated functionality seldom matched the actual programs submitted. Teachers should encourage students to develop programs that focus on simpler, more specific problems from the start. This would allow students to realistically develop what is proposed early in the documentation.

There were a few cases where students used the old Unit 2 syllabus to prepare their SBA. As a result, irrelevant information was submitted. Teachers should ensure they are using the syllabus marked **Effective for examinations from May/June 2009.**

RECOMMENDATIONS

Students are encouraged, as part of their examination technique, to read questions carefully before answering, and to respond with sufficient detail that is commensurate with the marks indicated for each question. Students need to answer questions in the context given instead of simply regurgitating notes. Past-paper questions should be carefully studied to understand how syllabus items can be tested.

Teachers need to prepare students adequately for the three modules of each unit. Algorithm Development, Computer Programming and Data Structures account for three of the six modules of the Computer Science syllabus so teachers need to cover these modules in more detail, scheduling as many practical sessions as possible. This can take the form of

- classroom discussions geared to solving specific problems or developing algorithms to solve given problems

- laboratory exercises involving the writing and testing of computer programs and experimentation with “buggy” programs

- using the Internet to download tutorials on programming and sample code to experiment with.

The poor performance by students on the Computer Architecture and Organization module can be handled by adequate coverage of the syllabus items. This module does not involve computer programming. Students need more practice with numerical problems such as those involving decimal to binary conversion and vice versa, as well as problems involving one’s and two’s complement.

With respect to the SBA, teachers need to provide better feedback to students. The SBA should be viewed as a process with well-defined milestones. As students submit work for these milestones, teachers need to provide adequate guidance, correcting errors made and indicating clearly what is required for the next milestone. It should also be mentioned that the SBA provides an invaluable opportunity to gain hands-on experience with the items in the Computer Science syllabus which ultimately leads to better performance on the written papers. Thus, the SBA should be viewed not as a goal in itself but as a framework for mastering the contents of the syllabus.

CARIBBEAN EXAMINATIONS COUNCIL

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION®

MAY/JUNE 2012

COMPUTER SCIENCE

GENERAL COMMENTS

This was the fourth year for which the revised syllabus was followed. There were three examination papers in each of Units 1 and 2, namely, Paper 01, Paper 02 and Paper 03. In each unit, Papers 01 and Paper 02 were examined externally by CXC while Paper 03, the School-Based Assessment, was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of multiple-choice questions that were designed to test candidates' breadth of coverage of the syllabus. Paper 02 consisted of essay-type questions that were designed to test candidates' depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade are 30 per cent, 50 per cent and 20 per cent, respectively.

DETAILED COMMENTS

UNIT 1 – FUNDAMENTALS OF COMPUTER SCIENCE

Paper 01 – Multiple Choice Questions

Performance on the 45 multiple-choice items on this paper produced a mean of 61 out of 90 with scores ranging between 20 and 90.

Paper 02 – Essay Questions

Section A – Computer Architecture and Organization

Question 1

This question tested candidates' knowledge of truth tables, logic gates, circuit diagrams, multiplexors, and the internal representation of data on a computer. It also required candidates to apply their understanding of a multiplexor to solve a simple problem.

For Part (a) (i), most candidates were awarded the marks for the input column combinations, however many of them failed to correctly work out the values for the output, F.

In Part (a) (ii), many candidates did not demonstrate a clear understanding of logic gates. Many of them confused the AND and OR gates, or did not understand how to place the NOT gates in the circuit design.
For Part (b) (i), some candidates were able to draw the block structure with four inputs and one output. However, most candidates did not include the two selection lines and sufficient labelling.

There were many non-responses for Part (b) (ii). This part was generally poorly answered. While some candidates were able to explain how a multiplexer works generally, they were unable to apply that knowledge to the specific use in the security system.

Part (c) (i) was generally answered well. Many candidates were able to correctly identify the largest and smallest integers as 7 and -7 respectively.

Many candidates who attempted Part (c) (ii) were able to recognize the sign bit as negative. Few were able to convert the binary string representing the mantissa to its decimal equivalent. Even fewer candidates were able to correctly identify the decimal equivalent of the 5-bit mantissa.

<u>Question 2</u>

This question tested candidates' knowledge of instruction set, instruction types, instruction formats, cache memory and storage devices. Overall, the question was answered poorly with a modal score of 6.

In Part (a) (i), candidates were able to define the *instruction set* but most of them failed to mention that the operation of the CPU is determined by the instructions it executes.

For Part (a) (ii), many candidates were unable to specifically identify the types of instructions (that is, data processing, data storage, data movement and control) however, accurate explanations were given for the aforementioned. In some instances, candidates misinterpreted the question and identified the instruction cycle (that is, fetch, decode, execute) rather than the types of instructions.

The responses for Part (a) (iii) were generally good but some candidates omitted the fact that a copy of frequently accessed RAM is stored in cache memory.

Part (b) (i) was well done as the majority of candidates were able to differentiate between the opcode and the operand of an instruction.

Part (b) (ii) was answered poorly. Many candidates gave no response. From the responses given, a few candidates were able to identify that the result of the operation of a one-address instruction format is stored in the accumulator. Candidates were also able to identify that there were two operands in a two-address instruction format.

The majority of candidates misinterpreted Part (c) (i). Very few candidates were able to differentiate between the access method of a hard disk (moving arm over a spinning disk) and RAM (electrical pathways). They were however able to correctly state that RAM had a faster access speed than a hard disk.

Part (c) (ii) was well done by the majority of candidates.

Section B – Problem Solving with Computers

<u>Question 3</u>

This question tested candidates' knowledge of algorithms, iterations within algorithms, and writing an algorithm based on the narrative of a given problem.

Most candidates scored between 13 and 17 for this question.

Part (a) tested candidates' knowledge of the role of algorithms in problem solving. All of the candidates attempted this part and were able to identify the role of algorithms in the problem solving process. Some candidates, however, were giving the definition of an algorithm instead of its role.

Most candidates attempted Parts (b) (i) and (ii) and showed some knowledge of iterations. Many of them could not justify their responses as they related to which algorithm illustrated bounded iteration and which illustrated unbounded iteration. Few candidates were able to differentiate between bounded and unbounded iterations.

Part (c) tested candidates' knowledge of the use of sequencing, selection and iteration in programming. Candidates did not understand the need to initialize the variables needed for the cumulated totals. Some candidates used a WHILE instead of the FOR statement but were unable to give accurate limits for the condition. Almost all candidates used the IF statement to select the correct option, however the condition was not clearly written. The calculation for the cumulative totals for each colour was fairly well done in most instances. The totalwas calculated and printed well by most candidates.

Part (d) tested candidates' knowledge of the use of iteration and selection in programming. Most candidates did not understand how to intialize variables and as result they scored zero. They also did not use the limits well in the WHILE loop and failed to increment the counter variable. Candidates did not use the modulus well, instead they simply divided in most instances.

Question 4

This question tested candidates' ability to represent an algorithm as a flow chart. It also examined their skills in tracing through an algorithm to determine its output.

Part (a) was fairly well done. Most candidates gave good responses. However, some of them drew incorrect symbols to represent initialization of variables. Some were uncertain about how to represent a loop in the flow chart. In a few cases, programming code was included with the symbols. The flow of logic was also not clearly shown for in many instances arrows were left out. More importance should be placed on constructing flow charts and candidates should make sure that they understand the purpose of each symbol used in a flow chart.

Part (b) was generally poorly answered. Many candidates were able to list the values of the variable 'sum' and output the answer, but they did not state what would be printed by the algorithm as was required. Additionally, candidates do not appear to understand how a condition is tested in order to terminate a loop. Candidates continued to test the loop even after the value 5 was entered, thereby generating output for the value 4 in the test data. While they recognized that the values of 5 and 7 in the test data would not be executed by the loop, they did not appear to understand that the value 5 would have ended the loop.

Part (c) was also fairly well done. Most candidates were able to trace through the algorithm. However, many of them were unable to produce the triangular shape consisting of $, + and &. In addition, there was some confusion with the use of "println", with some candidates placing the cursor on a new line and then placing the output on the new line. In general, most candidates who attempted scored at least five marks.

Responses to Parts (b) and (c) indicate that more emphasis needs to be paid to the tracing of algorithms, with special attention being given to the behaviour of loops.

Section C – Programming

Question 5

This question tested candidates' ability to write simple functions; manipulate files, create, read from, and write to files; and their knowledge of the stages in the translation process.

Part (a) was fairly well done. Candidates were asked to describe three of the stages of the program translation process. Some candidates were able to fully answer this part while others were at least able to correctly identify three stages. However, several candidates listed either the stages in the problem-solving process (problem definition, analysis, etc.) or steps in the instruction cycle (fetch, decode, transfer, etc.).

Part (b) was generally well done. Candidates were required to write a C function to calculate 2^n given n. Many candidates were able to obtain the majority of marks allocated but failed to state the relevant assumptions.

Part (c) was poorly done. Candidates were required to write a C function to write data to a file and then read the data stored, manipulate it and display the result. Most candidates were only able to establish the file pointer and open the file in the correct mode. Many were unable to properly form the loop to write to or read from the file.

Question 6

This question tested candidates' understanding of what constitutes good programming style. Candidates were also required to write C code to create and manipulate records consisting of different data types. The question also tested candidates' ability to take a specification of a problem and write a C program using an array which achieves the desired functionality. The key component of the solution was the *if-then-else* construct.

Part (a) tested the basic understanding of good programming techniques which candidates should have learnt at the CSEC level. The fact that so few candidates were able to obtain full marks for this question clearly indicates a lack of knowledge and understanding of the basic concepts in programming. While many candidates were able to identify three aspects of good programming style, few were able to offer a fair explanation as to why each was indicative of good programming.

Again being able to recognize what the question is asking in an important examination strategy that candidates seemed to lack. Candidates need to pay attention to key operational words which indicate the level of skill expected. Since candidates were asked to explain, simply identifying three ways would not gain them full marks.

Those who answered the question were able to identify the use of indentation, documentation, modularization, use of white spaces and meaningful variable names as techniques of good programming.
Part (b) tested candidates' ability to declare a record structure in C and to store and manipulate data in that structure.

The majority of candidates were unable to make the distinction between a record declaration tested in Part (b) (i) and the variable declaration tested in Part (b) (ii). As a result, few candidates were able to obtain full marks for this part.

Most candidates demonstrated knowledge of the *Struct* declaration in C but failed to use proper punctuation – semicolon and curly brackets.

Declaring variables for record structures and storing data in those variables also proved to be challenging for most candidates. Most candidates declared the variables as type *int* or *char* rather than the record structure *productRec.*

Part (b) (iv) required candidates to exchange the values stored in two records. The logic to perform this operation should have been acquired at the CSEC level. Again most candidates demonstrated a lack of understanding of this logic. As a result, this part of the question was not answered by most candidates or the answer written was not the most efficient code. Only a few candidates, therefore, were able to obtain maximum marks for this part.

Most candidates who answered the question copied the data 'field to field' rather than 'record to record.' This of course cost them valuable time in the exam and resulted in code that was long and cumbersome rather than simple and elegant.

For Part (c), many candidates were able to write C code that produced correct output. The efficiency of the solutions provided, however, needs to be improved. Very few candidates demonstrated an understanding of how to test multiple conditions in one IF statement. Instead most produced a nested-If statement.

While most candidates met the requirements for printing the number of occurrences of the vowel 'A' and the presence of a given vowel, few included code which tested for the absence of a vowel.

Paper 03 – School-Based Assessment (SBA)

UNIT 1

Students must be commended for their creativity when choosing an SBA project. Although most topics were recycled, students were able to be innovative in their delivery.

Many students were able to accurately describe the problem complete with the description of the current system and to give examples of what takes place when the problems arise. Some projects only included what activities the proposed system would perform without completely describing the current system and the problems which we are to be solved using software.

Narratives were fairly well written in most samples. However, some samples did not provide a completely accurate description of the algorithm. Pseudocode algorithms were generally well done. Samples which included flow charts in some instances incorrectly used system flow chart symbols and in others, diagrams were poorly presented and were difficult to follow. In the many samples using pseudocode algorithms, some were clearly modified copies of the programming source code. Students should be encouraged to develop their algorithms independent of the programming code.

It was quite evident that students were comfortable with procedural C programming language. This year's programs were logically written and properly decomposed. Nevertheless, some students disregarded the SBA requirement of using procedural C and used C++. A few samples did not make use of the key data structures (struct, files and arrays). Students must remember to print their source code directly from the compiler as transferring to a word processor changes the spacing of the code. Generally, most students did not include adequate comments at key areas of their code.

This year, the majority of samples provided a suitable range of test data, but a few students did not include all four testing criteria (normal, extreme, erroneous and incomplete) in their test plan. Test results must include actual screen shots of the working program and testing must be done using the test data outlined in the test plan.

Generally, this year's SBA projects were well presented. Teachers need to ensure that students use the headings outlined in the *Criteria For Marking Internal Assessment Project* in the syllabus, as well as follow the order in which these heading occurs. They should also ensure that students check that the numbering of their table of contents corresponds to the numbering in the document.

UNIT 2 – FURTHER TOPICS IN COMPUTER SCIENCE

Paper 01 – Multiple Choice

The performance on the 45 multiple-choice items on this paper produced a mean of 57 out of 90 with scores ranging between 14 and 90.

Paper 02 – Essay Questions

Section A – Data Structures

Question 1

This question tested candidates' knowledge of abstract data types and their implementation. It also examined candidates' knowledge of manipulating abstract data types to achieve a desired outcome. The question required candidates to manipulate a Stack ADT, a Linked List ADT, and a Queue ADT. It was attempted by more than 95 per cent of the candidates, however approximately 20 per cent gave satisfactory responses.
Part (a) (i) was poorly done. Candidates identified operations used with the Stack ADT (push, pop) but they did not answer the question by stating the difference between the Stack ADT and the C implementation of the Stack.

Part (a) (ii) was generally poorly done. Weaker candidates did not know the meaning of the word *variable* and listed *operations push and pop* in their answers.

Part (a) (iii) was poorly done. Candidates knew how to use the operations but could not write C code to implement them. Some candidates mixed up the conditions for underflow and overflow.

Part (b) was, surprisingly, poorly done. Most candidates identified the top and end of the linked list but placed the elements in the same order as given (indicating 43 as the top and 25 as the end)

Part (c) was generally well done. Candidates ignored the last statement in the question and used top, rear, front and other variables to write the algorithm.

Question 2

This question required that candidates show an understanding of how to manipulate a one-dimensional array structure using the C programming language. The question also tested candidates' knowledge of searching and sorting algorithms.

Part (a) required candidates to write C programming code in their responses, however many candidates provided pseudocode solutions. Generally, candidates knew that they had to use a loop to traverse the array; however, the condition that determined if the mark was within the range given was either poorly done or left out.

Part (b) required candidates to describe how the binary search algorithm may be used to locate an item that was in the list and one that was not in the list. Most candidates understood the general concept of the algorithm but there were two common mistakes. Firstly, when locating the 'middle' location the majority of candidates 'rounded up' instead of truncating and using the integer section, that is given 4.5 candidates used 5 instead of 4. The other common error occurred in the final 'divide and conquer' for an item that was not in the list. Candidates were unable to give an accurate description that determines when the item was not present.

Part (c) required candidates to draw an array after three passes of the selection sort algorithm. This part was fairly well done. However, a noticeable number of candidates sorted the entire array.

Section B – Software Engineering

Question 3

This question required candidates to construct a level-0 data flow diagram (DFD), and to demonstrate an understanding of the evolutionary and waterfall approaches to software development.

For Part (a), candidates were required to draw a level-0 DFD to depict the scenario of a registration system in a university. The DFD was poorly developed with many candidates confusing a DFD with an entity-relationship diagram (ERD). Many candidates did not use the correct symbols for the process, file and entity. Some candidates gave a context diagram rather than a more detailed diagram. Many candidates had DFD with data flows not labelled or labelled but with the direction of flow not shown. Exercises are required to help candidates improve on the drawing of DFDs.

Part (b) was poorly answered by candidates. Many candidates seemed to have studied the waterfall approach, and gave advantages and disadvantages of that approach.

In Part (c), candidates were required to describe four phases in the waterfall approach to software development (that is, analysis, design, coding and testing, maintenance). This part was generally well done. Most candidates who attempted this question were able to describe four phases.

Part (d) was satisfactorily answered by most candidates. Many cnadidates were able to give one reason for the involvement of the user in the software development process.

Question 4

This question tested candidates' knowledge of entity relationship models and their ability to draw the model from a given narrative of a system. The question also tested the candidates' knowledge of tests involved in binary search.

For Part (a) (i), most candidates were able to correctly identify the components of an entity-relationship diagram but in many cases they were not able to properly describe them, particularly the entity component.

In Part (a) (ii), candidates were generally able to draw the correct entity with only very few using the wrong symbol. Most candidates were also able to name the correct relationships. However, many were not able to represent the correct relationship cardinalities.

It was noted that with the exception of a few candidates those who used the Chen notations (1:1, 1:m, m:n) did not know the correct symbols for many to many , that is, m:n.

For Part (b), a large number of candidates were not able to describe a suitable test. Many described various aspects of unit testing. In cases where candidates described suitable tests for the binary search, most were only able to describe two of the three tests. Very few candidates were able to describe a suitable third test.

Section C – Operating Systems and Computer Networks

Question 5

This question tested candidates' knowledge of networking concepts such as network transmission media, network configurations, connectivity devices, transmission standards, and the role of the open systems interconnection (OSI) model in network communication. The least marks were scored in Part (d) wireless and Part (f) the OSI model.

From all indications, this section of the syllabus was not thoroughly covered as evidenced by the quality of candidates' answers.

Part (a) dealt with transmission media. Despite an attempt by most candidates to answer this question, it was not answered correctly. Many candidates were unable to adequately describe the characteristic of each for example, physical description, transmission capacity and mode of data transmission.

Part (b) tested candidates' knowledge of the ring topology. Most candidates who attempted this question had some knowledge of this topology. However, they could not represent it diagrammatically, for example, many drew a star topology instead of a ring when outlining the before and after situations, and the solution in their response.

This question also tested the candidates' ability to troubleshoot this topology. Most candidates changed the topology to a 'star' instead of maintaining the ring topology with the addition of a device or simply removing the malfunctioning computer to maintain the topology. Many candidates indicated the use of a hub or switch but some indicated router or gateway which showed that they did not fully understand the uses of these devices. There were those who converted the ring to a fibre distributed data interface (FDDI) which was not an appropriate solution.

In Part (c), only a few candidates were able to describe the role of a hub. They showed some knowledge of the device but lacked understanding of its purpose in a network. Diagrams were not well done, there were many missing labels.

Part (d) tested candidates' knowledge of wireless networking. The question was poorly done and in many cases was not attempted. Mainly, candidates did not know that IEEE802.11a is the established standard for wireless networks. Many candidates had difficulty identifying the devices needed and the transmission medium used in wireless networks. The diagrams were in many cases omitted or totally incorrect.

Part (e) which tested candidates' knowledge of firewalls was not well done, with most scoring one mark out of the three marks. Candidates only outlined the role, without including the aspects of security, hardware and/or software solutions, and blocking of IP addresses.

Part (f) assessed candidates' knowledge of the layers of the OSI and their corresponding functions. Though it was well attempted, knowledge of the names of the layer, order of Layers and purpose of each layer was lacking overall. The key to this question was to identify layers 1-5 only; Layer 1 being the physical layer and 5 the session layer. A few candidates labelled the application layer as Layer 1, which made the answer partially incorrect.

<u>Question 6</u>

This question tested candidates' understanding of various operating system concepts−deadlock, interrupt, process states, process control block, and user interfaces.

For Part (a), candidates generally responded well, with most providing a 'wait for' graph representing deadlock along with a correct description of what causes deadlock. The outcome however, stating that both processes must wait indefinitely, was often omitted.

In Part (b), most candidates provided at least a partially correct answer for how an interrupt is handled.

Most candidates who attempted Part (c) did not always define the process states by name, that is, *ready, running* and *blocked*, but gave partial descriptions for the states.

Part (d), was poorly done, with many candidates unable to correctly identify the components of a process control block. Some candidates stated components such as the program counter, accounting information and process state information.

Part (e), was well done. Only a few candidates had difficulty identifying an advantage of a menu interface over a command line interface.

RECOMMENDATIONS

Teachers should ensure that a thorough explanation of the printed circuit board (PCB) and its components be provided to students as this was the area with the highest omissions and margins of error for Question 6.

Teachers should also ensure that whilst they use visual aids to help students understand topics covered in class, they should make students aware that only computer related diagrams are acceptable and they should not provide those concept aids as responses to questions.

Paper 03 – School-Based Assessment (SBA)

UNIT 2

Students are reminded of the following:

- All programming language code should be printed from the C compiler (not from word processor)
- Examiners can only mark programming code from printed output and, therefore, to show that the program works, students need to print screen shots of the program in the testing phase and should not send soft copies.
- Only one form of algorithm is required that is either
    - Pseudocode OR
    - Flow chart

- Students are not required to describe the results of the analysis techniques.

Marking Criteria

<u>Definition of Problem</u>
Students were required to state the shortcomings of the existing system, and generally this section was well done by most of them. However, some students failed to identify a problem in the problem statement and therefore full marks were not awarded.

<u>Techniques of Analysis</u>
Students were required to identify techniques of data collection and analysis. Most of them were able to state the various techniques of data collection; however, they must be able to justify the reason for their selection as well as show proof of analysis. For example questionnaires and/or interview transcripts should be included in an appendix.

<u>Context Level Diagram</u>
In some instances, incorrect symbols were used for entities and processes. Some data flows were not labelled, and data store/file were incorrectly included in the context level diagram.

<u>Level 1 Diagram</u>
Students were required to give a complete and accurate diagram with all relevant processes, data flows and major data stores; however, some of them had links between files and entities which is incorrect.

<u>ER Diagram</u>
Students were required to give a complete and accurate diagram of all relevant entities and relationships. This was generally well done by most students; however, some of the cardinalities and also the description of the relationships were not done correctly.

<u>Functional and Non-Functional Requirements</u>
Students were required to give a complete and accurate description of all requirements. This was done well by some of them; however, some students sometimes confused or misinterpreted functional and non-functional requirements. Some students described project limitations for non-functional requirements.

Design Specification

Students were required to give a complete and accurate description of:

- System structuring
- User interface
- Report design
- Algorithm design
- Appropriate data structures

Some students did most of the design specifications, however:

- some of them went into too much detail in the systems structure by including modules or branches that were not necessary
- some of them did not name or justify the type of user interface they used
- some of them had no evidence of report design but were given marks by teacher
- some of them did not produce any algorithm.
- some of them were not awarded any marks by the teacher for appropriate data structure even though the program clearly showed the use of Struct, Array, Queues and Files declaration and manipulations. This suggests that either the teachers did not know or needed clarification on what to look for in this section.

Coding and Testing

Students were required to produce a complete and accurate program with:

- code that achieves functionalities such as documentations, outputs, usability and reporting
- code that corresponds to the design specification
- a test plan with exhaustive data set, (including unit testing, integrated testing and systems testing).

Most students produced programs that achieved good functionality. However, some of them did not produce proper test plans to cover unit testing, integrated testing and systems testing. Some students only produced test results and screen shots but NO test plan.

RECOMMENDATION

Teachers should ensure that students are fully prepared for the examinations in both units. The poor performance in some modules of the syllabus indicates that more time needs to be spent on some areas.

CARIBBEAN EXAMINATIONS COUNCIL

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION®
MAY/JUNE 2013

COMPUTER SCIENCE

GENERAL COMMENTS

This was the fifth year in which the revised syllabus was examined. There were three examination papers in each of Units 1 and 2, namely, Paper 01, Paper 02 and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the School-Based Assessment, was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of multiple-choice questions that were designed to test candidates' breadth of coverage of the syllabus. Paper 02 consisted of essay-type questions that were designed to test candidates' depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade are 30 per cent, 50 per cent, and 20 per cent, respectively.

Approximately 93 per cent of the candidates obtained Grades I─V in Unit 1 and approximately 93 per cent of the candidates in Unit 2 obtained Grades I─V. Overall, there is still need for improvement in the quality of responses for the programming questions in both units.

DETAILED COMMENTS

UNIT 1 – FUNDAMENTALS OF COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of approximately 60.51 out of 90, standard deviation of 13.87 and scores ranging from 16 to 90.

Paper 02 – Essay Questions

Section A – Computer Architecture and Organization

Question 1

This question tested candidates' knowledge and understanding of logic gates, circuits and binary representation of negative integers and floating point numbers.
The question was generally well done with the majority of the candidates obtaining more than 50 per cent of the marks available for this question.

Part (a) (i) was satisfactorily done. A few candidates either attempted a circuit diagram instead of a block diagram or did not include selector lines.

Part (a) (ii) was fairly well done. Many candidates were able to partially explain how the multiplexer operates but did not go on to disclose the values of selector lines used to obtain the signals in l_1 and l_3 respectively.

Part (a) (iii) was generally well done. The majority of candidates was able to state one property and one use of flip flops.

Part (b) (i) was well done. A few candidates were unable to convert 5 to binary.

Part (b) (ii) was fairly well done. However, a number of candidates found the one's complement of the sign and magnitude representation from Part b (i).

Part (b) (iii) was also fairly well done because candidates went on to find the two's complement using the sign and magnitude value from Part (a) (i).

Part (c) was generally poorly done, with a significant number of candidates being unable to obtain any of the three marks available. Many candidates simply converted the entire representation to decimal to obtain a value of $165_{10.}$

Part (d) was exceptionally well done with the majority of candidates being able to obtain full marks for drawing the truth tables for the NOT, AND and OR logic gates respectively. Weaker candidates did not correctly map each truth table to its corresponding logic gate.

Common Misconceptions

- Several candidates stated that a property of a flip flop is the ability to store one bit of data when this is in fact a use of flip flops.
- Many candidates identified examples of bistable devices such as light switches, alarm systems etc. instead of stating a use of a flip flop.

Question 2

This question mainly tested candidates' knowledge of port connectivity, instruction sets and direct addressing.

Part (a) was done relatively well. However, many candidates either did not explain the term *port connectivity* as it related to the given situation or misunderstood the term and explained it in terms of the ports malfunctioning due to rust, dust and general physical deterioration over time.

Candidates answered Part (b) well. Some, however, simply stated what the acronyms meant, as a way of differentiating between the items given.

For Part (c), most candidates were able to arrange the given computers in order of size from smallest to largest.

Part (d) was poorly answered overall. In Part (d) (i), most candidates were able to explain what is meant by the *instruction set of a computer*.

In Part (d) (ii), many candidates were unable to accurately identify the three types of instructions and often confused the examples given.

In Part (d) (iii), most candidates were unable to explain the term *direct addressing* and many explained indirect addressing instead.

Part (e) was also poorly answered overall, with candidates having problems adequately explaining how the fetch, decode and execute cycles work in a computer. The majority of candidates misused the terms 'data' and 'information' in their explanation. They also reused the terms *fetch, decode* and *execute*, to explain the cycle on the computer.

Section B – Problem Solving with Computers

Question 3

This question tested candidates' ability to

- identify and describe the stages in the problem-solving process
- generate algorithms based on a given scenario
- construct a flow chart based on a scenario

While the majority of the candidates was able to confidently attempt the question, there were a few who demonstrated little knowledge of the subject content.

Part (a) was generally well done. A few candidates were only able to identify the problem-solving phase but could not give an appropriate description.

Part (b) asked candidates to write an algorithm based on the given scenario. This question could have been answered by using pseudocode, flow chart or narrative as examples of algorithms. Many candidates wrote pseudocode and it was generally well done. A few candidates used a flow chart or a narrative. The majority of candidates was able to write the algorithm using the constructs correctly and appropriately. Only a few candidates wrote the algorithm using code-like statements and calculated the cost of the calls incorrectly.

Part (c) required candidates to construct a flow chart based on the given scenario. It was generally well done by the majority of the candidates. However, a few were not able to correctly and appropriately use the flow chart symbols. For example, some candidates used the input/output symbol for processing and vice versa.

Recommendations

Greater focus must be given to the following areas:

- Correct use of technical terms and giving appropriate descriptions of each
- Correct flow charting principles. For example,
    - Correct and appropriate use of symbols
    - Correct and appropriate representation of constructs such as loops (repetition)

<u>Question 4</u>

This question tested candidates' ability to trace through an algorithm to determine its output. It also examined their ability to write an algorithm for a particular case.

Part (a) was fairly well done. Most candidates attempted the question and were able to trace through the algorithm. However, few candidates produced the correct pattern. Some candidates are still having problems distinguishing between the *write* and *writeln* statements. The majority of candidates scored between 5 and 6 out of a maximum of 11 marks for this part.

Part (b) was generally well done. Many candidates were able to produce an algorithm using either the *while* or the *for* construct.

Part (c) was fairly well done. While most candidates were able to provide three properties of well-designed algorithms, the difficulty came in accompanying them with examples in each case. This suggests a lack of thorough understanding of properties of algorithms.

Section C – Programming

Question 5

This question tested candidates' ability to unite a function swap and also write a C program based on particular instructions.

Part (a) was generally well done. Candidates were required to explain two differences between compilers and interpreters. Most candidates were able to properly explain at least one difference.

Part (b) was well done overall. Candidates showed good understanding of the need to introduce a temporary variable in order to successfully execute the *swap* function.

Part (c) was generally well done. Most candidates were able to correctly write statements to compute and output the integers in reverse order, along with the sum and product. However, in some cases, the actual C code was not syntactically correct.

Part (d) was not well done. This question required candidates to write a recursive or non-recursive program to output the factorial of an integer. Responses suggest that more practice is needed on questions of this type.

Question 6

This question tested candidates' knowledge of different programming paradigms and their ability to write C code to manipulate records consisting of different data types.

Part (a) tested the basic knowledge of different programming paradigms. Most candidates did not demonstrate understanding of the term *paradigm*, and hence could not answer the question satisfactorily. Part (b) was attempted by most candidates. However, candidates gave differences between a mobile device and a desktop instead of a reason for the difference in language implementation.

Part (c) was attempted by most candidates and was fairly well done. Most candidates correctly provided the first four lines of the output. There seemed to be some difficulty coming to terms with the variable *j* as a control variable in the outer *for* loop, and then being changed internally by the *while* construct. Many candidates were able to score four out of six marks for this part.

Performance on Part (d) was satisfactory. For Part (d) (i), some candidates were unable to clearly define a structure with fields. Many candidates were able to place values into the proper variables for Part (d) (ii) but some did not use meaningful variable names based on the context of the question. The discount for Part (d) (iii) was not properly applied generally because candidates were unable to access the structure variables properly. Part (d) (iv) was fairly well done; some candidates did not use a temporary variable to handle the swapping process.

Paper 03 – School- Based Assessment (SBA)

UNIT 1

Students must be commended for their creativity in choosing an SBA project. Although most topics were recycled, students were able to be innovative in their delivery.

Definition of Problem

Many students were able to accurately describe the problem complete with the description of the current system, giving examples of what takes place when the problems arise. However, some students' projects only included partial definitions of activities of the proposed system that they intended to create and did not completely describe the current system and the problems which are to be solved using software.

Narrative and Flow Charts or Pseudocode

Narratives were fairly well written in most samples. However, some samples did not provide:

- A completely accurate description of the algorithm
- A description of the step by step procedure that will solve the problem as required by an algorithm

Pseudocode algorithms were generally well done. Samples which included flow charts in some instances incorrectly used some symbols, for example, system flow chart symbols and in others cases, diagrams were poorly presented, had symbols with no labels (such as yes/no for the decision symbol), no connector symbols to connect diagrams to the next page and therefore were difficult to follow. In the samples using pseudocode algorithms, some were clearly modified copies of the programming source code, with a few being an almost identical copy of the source code. Students should be encouraged to develop their algorithms independent of the programming code.

Coding of Program

Students' projects demonstrated that they were comfortable with procedural C programming language. This was quite evident in the various projects. Most programs were logically written and properly decomposed. Nevertheless, some students' sample

- did not comprise functions as independent units and were still awarded full marks by the teacher
- had few data structure in the program
- only had evidence of sequence and selection but no loops for the concept of structured programming
- did not make use of the key data structures (struct, files and arrays)

Students are reminded to print their source code directly from the compiler as transferring to a word processor changes the spacing of the code. This may lead to students losing marks for appropriate programming style and documentation. Generally, most students did not include adequate comments at key areas of their code and a few were not properly indented. However, this section was done well by most students.

Evidence that Code Matches Algorithm

Most students were able to obtain full marks on this section.

Evidence of File Manipulation

Students were required to provide code that include file and show evidence of file manipulation (open file, write to file, read from file, append file and close file). Most students demonstrated evidence of this. However, some students were awarded marks by the teacher even though there was no evidence of file manipulation and in some cases the use of file was non-existent.

Testing and Presentation

This year the majority of samples provided a suitable range of test data, but a few students:

- Did not have a test plan, but had screen shots only
- Did not include all four testing criteria (normal, extreme, erroneous and incomplete) in their test plan
- Had test results but did not have a clear test plan
- Had test which results did not include actual screen shots of the working program and testing must be done using the test data outlined in the test plan

Generally, this year's SBA projects were well presented.

RECOMMENDATION

Teachers need to ensure that the students use the headings outlined in the Criteria for Marking Internal Assessment Project in the syllabus, as well as follow the order in which these heading occurs. They should also ensure that students check that the numbering of their table of contents corresponds to the numbering in the document and that they provide information in a logical way using correct grammar and appropriate jargon at all times in the presentation of their projects.

UNIT 2 – FURTHER TOPICS IN COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of 63.35 out of 90, standard deviation of 11.78 and scores ranging from 25 to 88.

Paper 02 – Essay Questions

Section A – Data Structures

Question 1

The question tested the stack ADT, linked lists and associated operations. It was attempted by most candidates. Overall, candidates performed better in Part (a) compared with Part (b).

In Part (a) (i), candidates were expected to declare the stack variables using C syntax. Some syntactic errors were found in responses. Some candidates declared the stack with an incorrect type. Overall, many candidates generally gave proper declarations.

Parts (a) (ii) and (iii) required functions for *push* and *pop* respectively. Some candidates, although their *push* and *pop* functions were not completely correct, were able to increment and decrement the *top* variable properly.

In Part (b) (i), the linked list diagram needed to clearly illustrate the head/start and null positions, the composition of a simple node and the direction of the link. Most candidates showed an understanding of the linked list structure. In Part (b) (ii), candidates were expected to explain the actions involved when inserting a node to the front and back of a linked list. Many candidates did not explain how the pointer references would change during the insertion of nodes. Overall, more practice is needed with these types of questions.

Question 2

This question tested candidates' knowledge of sorting and searching concepts.

Part (a) tested candidates' ability to interpret, explain and use a bubble sort function written in C. Most candidates recognized that a swap was being performed in the inner *for loop*. They were also able to determine the content of the array after one stage of the bubble sort. Some candidates had trouble explaining the purpose of the *for loops* in the bubble sort.

Part (b) tested candidates' ability to sort a given array using selection sort. Responses were satisfactory but some candidates did not know what a *pass* was.

Part (c) tested candidates' ability to describe the principle of the binary search. Most candidates knew the basic concept of the binary search. The idea of a divide and conquer approach of binary search was familiar to many candidates. Some confused binary search with linear search.

Part (d) tested candidates' understanding of the operation of a linear and binary search. Most candidates were able to determine the number of comparisons needed to determine if a key was not present using both binary search and linear search. Although the basic concept of the binary search was well understood, most candidates did not have a detailed understanding of its operations. Therefore, most were not able to correctly trace a search.

Section B – Software Engineering

Question 3

Part (a) (i) was fairly well done. Most candidates were able to link prototyping with building a system or software testing and refining. However, some candidates failed to mention that the end-user was responsible for testing and providing feedback. Also, a number of candidates incorrectly described beta testing instead of prototyping.

Part (a) (ii) was generally well answered. A substantial number of candidates scored 3–4 marks out of a maximum of 4. Candidates were able to describe the iterative process of building software, allowing users to test and provide feedback and then refining the software, as beneficial to creating a system that meets the needs of the user.

In Part (a) (iii), many candidates simply stated the weakness of prototyping as being time consuming and expensive. While some candidates described how time consuming the iterative process of evolutionary prototyping is and how expensive it is to constantly build prototypes, many candidates did not provide any justification for their responses and lost marks as a result.

For Part (b), some candidates did very well to create an entity relationship diagram. Most candidates were able to identify and draw the entities correctly. Candidates were also generally able to identify the relationships between the entities, with a few candidates having duplicate entities and relationships. Some of the errors candidates made included using improper notation for the entities and representation of cardinalities. Also, many candidates failed to properly represent the attributes and the primary key for Patient. Generally, candidates generally demonstrated an understanding of what entities are as well as the cardinalities and relationships between these entities but the ability to represent this understanding in a diagram using standard notation seemed to be an issue for some.

Many of the responses from candidates for Part (c) indicated that they did not understand what was being asked. Consequently, many candidates scored zero because most of them wrote about different forms of testing such as integration testing and abnormal testing. While some candidates were able to correctly identify two appropriate tests by searching for a key that is known to be in the array and a key that is not known to be in the array, many of these candidates failed to get the third test correct since they tested for an inappropriate data type (for example, string or chart) rather than testing an empty or null array.

Question 4

This question mainly tested candidates' knowledge of data flow diagrams (DFDs.) Part (a) was generally well done. It tested the traditional methods of determining the requirements of a new system and the advantages and disadvantages of each. Most candidates generally answered correctly by stating questionnaire, interview, etc. However, some candidates misinterpreted the question and gave answers related to a system development life cycle (for example, waterfall approach and prototyping).

Part (b) tested candidates' knowledge of DFDs. Most candidates provided good responses. Many candidates did not clearly state that an entity is a source of data *external* to the system. Many candidates also stated the purpose of a data flow which was not specifically asked for. This part of the question specifically asked for three separate diagrams for each of the symbols. It tested the drawing and the use of the symbols. Some candidates used one diagram but labelled it correctly. However, some candidates drew one diagram but did not label it so no marks could be awarded. In addition, some candidates drew separate diagrams but did not label them. Some candidates also gave the correct symbol but no data flow.

Many candidates gave a context diagram when the question clearly asked for a level 0 DFD. Candidates lost marks for the data flow in the diagram because they used verbs to describe the flow of data. Also, the warehouse was placed as an entity in some DFDs when in fact the warehouse was not an external entity.

For Part (c), few candidates were able to correctly distinguish between functional and non-functional requirements and give correct examples. Others gave correct definitions but the wrong examples and vice versa.

Overall, most candidates attempted all parts of this question.

Section C – Operating Systems and Computer Networks

Question 5

This question tested candidates' knowledge of networking concepts such as network transmission media, network configurations, connectivity devices, transmission standards, and the role of the OSI model in network communication.

Part (a) dealt with a peer-to-to peer network. Despite an attempt by most candidates to answer this question, it was generally not answered correctly. Many of them were unable to adequately mention that the roles can interchange.

Part (b) (i) tested candidates' knowledge of network security. Most candidates understood the concept very well and earned high marks.

Part (b) (ii) tested candidates' ability to explain the network issue of expandability. Most candidates understood that more devices or users would need to be added. However, some were not aware how to modify to the network.

Part (b) (iii) tested candidates` knowledge of interconnectivity of the network. Most candidates wrote about connecting within the network rather than with other networks. This part of the question was generally poorly done by candidates.

In Part (c), only a few candidates were able to list the correct order of the OSI layers. Many candidates listed layers starting from 7 to 3. Most candidates were unable to describe the role of three layers. They did not understand the role of the different layers.

Part (d) tested knowledge of a network topology. Most candidates drew and labelled the diagram correctly. However, some candidates had the server as the central device rather than the switch.

Question 6

This question tested candidates' knowledge of process management, in particular their knowledge of process states and process control blocks. It also examined their understanding of the different types of operation systems and device management.

Parts (a) and (b) were poorly done. Most candidates recognized state of process and process identification but could not identify the other components. Candidates could have stated that the process control black (PCB) contains important information about specific processes. Some components of the PCB are:

- The current state of the process, for example, ready runway
- Priority of process which CPU can use for scheduling
- Register information where data can be stored

In Part (c), most candidates provided a partially correct answer for a device driver. The full response expected was: *divide drivers contain instructions to control the hardware components that are linked to the operating system code used.*

Part (d) was well done with candidates giving satisfactory answers for types of operating systems.

Paper 03 – School- Based Assessment (SBA)

UNIT 2

Students performed very well, with many following the marking guidelines.

Definition of Problem

Students were required to state the shortcomings of the existing system, and generally this section was well done by most of them. However, some students failed to clearly describe the problem symptoms evident in the existing system. There was too much description of the current system but not enough explanations of the problems staff/clients face.

Techniques of Analysis

Students were required to identify techniques of data collection and analysis. Most of them were able to state the various techniques of data collection. However, students must be able to justify the reason for their selection as well as show proof of analysis. For example, questionnaires and/or interview transcripts should be included in an appendix. The proper usage of the techniques was not described in the context in which the problem was encountered.

Context Level Diagram

In some instances, incorrect symbols were used for entities and processes. Some data flows were not labelled, and data store/file were incorrectly included in the context level diagram.

Level 1 Diagram

Students were required to give a complete and accurate diagram with all relevant processes, data flows and major data stores; however, some of them had links between files and entities which were incorrect.

ER Diagram

Students were required to give a complete and accurate diagram of all relevant entities and relationships. This was generally done well by most of them; however, some of the cardinalities and also the description of the relationships were not done correctly.

Functional and Non-Functional Requirements

Students were required to give a complete and accurate description of all requirements. This was done well by some of them. However, some students confused or misinterpreted the difference between functional and non-functional requirements. Some of them described project limitations for non-functional requirements.

Design Specification

Students were required to give a complete and accurate description of:

- System structuring
- User interface
- Report design
- Algorithm design
- Appropriate data structures

Some students did most of the design specifications; however, some of them

- went into too much detail in the systems structure by including modules or branches that were not necessary
- did not name or justify the type of user interface they used
- had no evidence of report design but were given marks by teachers
- did not produce any algorithm
- were not awarded any marks by the teacher for appropriate data structure even though the program clearly showed the use of struct, array, queues and files declarations and manipulations. This suggests that either the teachers did not know or, needed clarification on what to look for in this section.

Coding and Testing

Students were required to produce a complete and accurate program with

- code that achieves functionalities such as documentations, outputs, usability and reporting
- code that corresponds to the design specification
- a test plan with exhaustive data set (including unit testing, integrated testing and systems testing).

Most students produced programs that achieved good functionality. However, some of them did not produce proper test plans to cover unit testing, integrated testing and systems testing.

RECOMMENDATIONS

- Teachers should ensure that students are fully prepared for the examinations in both units. The poor performance on some modules of the syllabus indicates that more time needs to be spent on these areas.

- All programming language code should be printed from the C compiler (not from word processor)

- Examiners can only mark programming code from printed output and therefore, to show that the program works, students need to print screen shots of the program in the testing phase and should not send soft copies.

- Only *one* form of algorithm is required that is, either
  - Pseudocode OR
  - Flowchart

- Students are not required to describe the results of the analysis techniques.

C A R I B B E A N   E X A M I N A T I O N S   C O U N C I L

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION®
MAY/JUNE 2014

COMPUTER SCIENCE

GENERAL COMMENTS

This was the sixth year in which the revised syllabus was examined. There were three examination papers in each of Units 1 and 2, namely, Paper 01, Paper 02 and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the School-Based Assessment (SBA), was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of multiple-choice questions which were designed to test candidates' breadth of coverage of the syllabus. Paper 02 consisted of essay-type questions that were designed to test candidates' depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade are 30 per cent, 50 per cent, and 20 per cent, respectively.

Approximately 93 per cent of candidates obtained Grades I─V in Unit 1 and approximately 92 per cent of candidates in Unit 2 obtained Grades I─V. Overall, there is still need for improvement in the quality of responses for the programming questions in both units.

DETAILED COMMENTS

UNIT 1 – FUNDAMENTALS OF COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of approximately 61.20 out of 90, standard deviation of 14.64 and scores ranging from 24 to 90.

Paper 02 – Essay Questions

Section A – Computer Architecture and Organization

Question 1

Part (a) tested candidates' ability to draw a clearly labelled block diagram of a 2-to-1 line multiplexer. Most candidates attempted this question gaining at least three of the five marks. Many candidates lost marks mostly through failing to realize that a 2-to-1 line multiplexer has one select line. Candidates also lost marks via omission of the select line and improper labelling.

Part (b) required candidates to define the term *logic gate*. Most candidates wrote that logic gates take inputs and provide outputs according to their logical rules. Candidates did, however, lose marks by failing to note that logic gates are elementary building blocks/components of digital circuits.

Part (c) provided a circuit diagram and asked candidates to provide the truth table associated with the diagram. This part was fairly well done. Candidates are however encouraged to maintain order when displaying the input values for the truth table (for example, 0 0, 0 1, 1 0, 1 1).

Part (d) required candidates to indicate which of *decoder*, *flip-flop and multiplexor* are associated with a single input. Most candidates were able to identify at least one out of the two correct devices (*flip flop and multiplexor*). Candidates were not required to provide definitions for the structures identified.

For Part (e) (i), candidates were required to provide the number of digits that a 3-bit decoder would display. Many candidates lost marks by answering 6 (2 x 3). A decoder can provide 2” outputs where n is the amount of different inputs. Therefore the number of outputs would be 8.

For Part (e) (ii), candidates were required to examine the digital figure provided and list the segment letters that must be switched on to display the number 5. This part was well done. Candidates provided most or all of the segments needed. Some candidates provided the segments that should be off and were awarded no marks for their response.

For Part (f), the responses provided by candidates were mostly correct although some candidates included sign and magnitude alongside the complement process and added a sign bit before inverting the value. Candidates must be mindful of the number of bits required for the question as those whose final responses were not represented using 4 bits lost marks as a result.

Part (g) asked candidates to determine what decimal number was being represented by the given binary string. Many candidates did not seem to have knowledge of this process and some candidates simply treated the binary as an absolute value and converted it to its decimal equivalent. Candidates were expected to have displayed the component parts of the string showing *0 for the sign, 011 = 3 for the exponent* and *10110 for the mantissa*. Many candidates failed to recognize that the mantissa is a fraction and should be written as 0.10110.

<u>Question 2</u>

Part (a) tested candidates' ability to describe in the correct order, the three main activities in an instruction cycle. Candidates were able to correctly identify- in the correct order- the three main activities that take place in an instruction cycle as fetch, decode and execute. Many candidates attempted to discuss the activities outlined but gave vague explanations of each.

In Part (b) (i), candidates were required to give one similarity and one difference between RAM and hard disks. This part was well done.

Part (b) (ii) tested candidates' knowledge of registers used in the CPU of a computer. Few candidates were able to provide the correct responses to this part. Most neglected to thoroughly explain the reasons for including registers in the CPU.

For Part (c), almost all candidates attempted to describe situations where a Supercomputer, PDA and Mainframe can be used. However, many candidates gave definitions or explanations of these categories of computers rather than their uses.

Part (d) was very well done.

For Part (e), whilst the majority of candidates correctly identified port connectivity, speed and software incompatibility as reasons why current computers may not be able to work efficiently with new storage devices, very few adequately discussed these points resulting in their inability to obtain full marks.

Section B – Problem Solving with Computers

Question 3

This question tested candidates' ability to:

- Identify and give examples of the main control constructs used in programming.
- Generate an algorithm to select the favourite subjects of 100 students.
- Generate an algorithm to calculate the sum of the multiples of six and seven between 100 and 250.

For Part (a), the majority of candidates described the control constructs and gave an example of each. This part was fairly well done.

Part (b) was done well by the majority of candidates. A few candidates did not initialize the variables needed to perform the counting operation. Some candidates also used the selection construct improperly.

Part (c) proved challenging for some candidates. Those who did not complete the question had difficulty testing correctly for the multiples of the given numbers.

More attention should be focused on the following:

- correctly using the sequence, selection and iteration control constructs
- using functions such as modulus to test numbers for specific properties
- correctly selecting the limits of loops based on given scenarios

Question 4

Part (a) tested candidates' ability to write an algorithm using the iterative control construct to solve a given problem; Part (b) required candidates to represent the algorithm with the use of a flowchart and Part (c) tested candidates' ability to trace an algorithm to determine its output.

More than 90 per cent of the candidates attempted this question and gave satisfactory responses.

Part (a) was fairly well done. The weaker candidates confused the data in the example given as a solution to their algorithm. Some candidates used the iterative control construct improperly, using the "$" in the IF statement as well as failing to initialize the variable for the running total. Some candidates also answered the question using the C programming language rather than the construction of the algorithm.

Part (b) was attempted by most candidates and was generally well done. Some of the weaker candidates used the flowchart symbols incorrectly and some excluded the iterative statement in the flowchart.

Part (c) was also attempted by the majority of candidates and they performed very well. In most cases, they scored full marks. Some candidates did not change the value of the variable 'x' for the trace table which resulted in incorrect values.

Section C – Programming

Question 5

This question tested candidates' knowledge of the stages in the translation process of a program, and the purpose of the watch window in the C compiler. Candidates were also required to write a C function based on a given scenario and apply knowledge of FILE processing.

Part (a) was poorly done by most candidates. Candidates attempting this part seemed to be familiar with the lexical analysis stage only.

Part (b) was poorly done. Most candidates simply did not know the purpose of the watch window in the C compiler.

Part (c) was poorly done. Candidates seemed not to know the difference between a *main* function and a *programmer defined* function, hence many wrote their solution using main. Those candidates who did well were able to write a proper function header, sum the elements of an array and return its total. Candidates lost marks if they did not declare and initialize their variables.

Part (d) was poorly answered. Most candidates who attempted this part of the question knew how to create a FILE pointer. However, candidates had difficulty writing the appropriate structure of *fscanf*, *fprintf*, and *fopen* statements. Candidates seemed not to know when to use the different modes of opening a file, for example, (r, w, a, r+, w+, a+).

<u>Question 6</u>

This question was designed to test candidates' ability to:

- Explain the importance of good programming style.
- Interpret source code and utilize their knowledge to formulate the output of given code.
- Write a C program that tests their knowledge of program structure, variable declarations, input and output of variables, and their ability to formulate an equation to solve the addition of taxes on an existing price.
- Write C code that tests their ability to read and print numerical values after performing an arithmetic calculation to solve the average of those numbers.

Part (a) was generally well done. Some candidates failed to explain the importance of indentation clearly and lost marks. A few candidates confused the purpose of indentation and associated it with a benefit towards the computer or compiler.

Part (b) was well answered; only a few candidates did not elaborate their answers properly.

Part (c) was not clearly understood by many candidates; the majority failed to provide the actual output that was being requested. The majority of candidates took the approach of providing a trace table and some simply attempted to show some sort of calculation to arrive at their answers.

Most candidates failed to analyse the loops properly. Upon entering the *while* loop, candidates performed well in obtaining the result of the first stage but failed to re-enter the *for* loop with the appropriate data hence, obtaining wrong data thereafter.

Part (d) was generally well understood by most candidates but some failed to write correct syntax either in the form of program structure: missing out # or braces; variable declarations: declaring as *int*; reading of inputs: missing out the *&*; and in their output: not formatting their result as *%6.2f*. Some candidates failed to analyse the mathematical problem as most failed to divide the *tax* by 100.

Part (e) seemed to be unclear to most candidates as it pertains to the required result. Most candidates read variables properly and solved the result but failed to provide their answer as a valid average of type *float*. Some candidates failed to apply proper precedence of operations by omitting brackets and some even ended up dividing the sum of all numbers read by values such as 2 and 3 when the question clearly asked for 4 numbers. The majority of candidates failed to provide their result as a *float* data type.

Candidates are encouraged to pay close attention to the use of proper coding syntax when asked to write C programming code. Candidates are also encouraged to carefully read and analyse the questions to fully understand what is being requested of them.

Paper 03 – School-Based Assessment (SBA)

General Requirements

Students were expected to choose a problem for which a software solution was appropriate; develop and present the solution in a logical way using correct grammar and appropriate jargon at all times by doing the following:

- Create an algorithm for the solution using modules, sequencing, selection, assignments, and iteration (bounded and unbounded).
- Represent their algorithms using narrative format and also as either a flow chart or pseudo-code.
- Implement the algorithm in C using various data structures such as arrays, struct, strings and files, with no less than five functions as independent units.
- Ensure that the source code produced, matches the algorithm.
- Create a test plan with exhaustive data set, test and produce test results and appropriate error messages.

The following specific aspects of the project were assessed:

- Definition of problem;
- Narrative and flow charts or pseudocode;
- Coding of program;
- Testing and presentation;
- Communication of information.

For each component, the aim was to find the level of achievement reached by the student. It is recommended that teachers make the assessment criteria available to students at all times.

Definition of problem

Many students were able to accurately describe the problem complete with the description of the current system and giving background to the problem, the issue or problem, recommended solution and examples of what takes place when the problem arises. However, some students' projects only included partial definition of activities of the proposed system that they intended to create without completely describing the current system and the problems which are to be solved using software. In addition, some problems that were identified could not be solved by using a computer program. For example, filling out of a form is a time-consuming and error-prone process for both paper-based and computer-based systems. However, there are advantages of using computer-based forms that were not identified.

Narrative and flow charts or pseudocode

Narratives were fairly well written in most samples. However some samples:

- Provided only a partially correct description of the algorithm as a solution.
- Provided descriptions of an intended module or unit instead of describing the step by step procedure that would solve the problem as required by an algorithm.

Pseudocode algorithms were generally well done. Samples which included flowcharts in some instances incorrectly used some symbols. For example, system flowchart symbols were used. In others cases, diagrams were poorly presented, had symbols with no labels (such as yes/no for the decision symbol), had no connector symbols to connect diagrams to the next page. Charts were therefore difficult to follow. Most students produced samples using pseudo-code algorithms. Some samples were clearly modified copies of the programming source code, and a few almost identical copies of the source code. It is therefore recommended that in the future, students be encouraged to develop their algorithms independent of the programming code.

Coding of program

Students' projects demonstrated that they were comfortable with procedural C programming language. This was quite evident by the responses of the various projects. Most programs were logically written and properly decomposed. Nevertheless, some students:

- Did not use functions as independent units in their programs and yet some teachers still awarded full marks.
- Used too few data structures in their programs.
- Did not demonstrate appropriate use of the concept of structured programming.

Some students did not print their programs from the C compiler, instead choosing to make print screen copies of their projects. Students are therefore reminded to print their source code directly from the C compiler as transferring the code to a word processor often changes the code. For example, transferring to a word processor may cause adjustments to the spacing and cause strange symbols to occur in the code, making it difficult to read. This may also lead to students losing marks for inappropriate programming style and documentation. In addition, marks were lost because some students did not include adequate comments at key areas in their source code, and they used a poor indentation style that did not make the code easier to read. However, this section was generally done well by most students.

Evidence that code matches algorithm

Most students were able to obtain full marks on this section as this was well done.

Evidence of file manipulation

Students were required to provide code that included a file and show evidence of file manipulation that is, open file; write to file; read from file; append file and close file.

This section was well done. However, some students were awarded marks by the teacher even though there was no evidence of file manipulation and in some cases the use of files was non-existent or was declared but not used, and the teacher still awarded full marks. Projects that showed screenshots of the file data being added, edited and deleted were awarded full marks.

Testing and presentation

Students were required to prepare a test plan with an exhaustive data set (test data) to test their programs. Students were also expected to use the test data set to produce test results with normal input giving correct results, extreme input giving correct results and erroneous (abnormal) input giving appropriate error messages. The majority of samples moderated this year did not have a suitable range of test data. However, a few students:

- Did have a test plan and screen shots while others provided screen shots without any test plan.
- Did not include all testing criteria (normal, extreme and erroneous) in their test plan.
- Had test results but did not have a clear test plan.
- Had no test results but were awarded marks by teacher.

- Showed test results but did not include actual screen shots of the working program. Testing was not done using the test data outlined in the test plan.
- Tested their menus with the appropriate test data, but all input must be tested in the same way.

Generally, this year's SBA projects were well presented.

Recommendation

Teachers need to ensure that students use the headings as outlined in the syllabus. In addition, students should follow the order in which these heading occur in the Form CSCI 1-3. They should also ensure that students check that the numbering in the table of contents corresponds to the numbering in the document and that students provide information in a logical way using correct grammar and appropriate jargon at all times in the presentation of their projects.

UNIT 2 – FURTHER TOPICS IN COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of approximately 62.19 out of 90, standard deviation of 13.09 and scores ranging from 17 to 86.

Paper 02 – Essay Questions

Section A – Data Structures

Question 1

Part (a) tested candidates' knowledge of the creation and deletion of a *STACK* (*ADT*). The majority of candidates was able to correctly identify that a stack only needed to exist in order to execute the *Destroy Stack* function. The modal mark was two in this section.

For Part (b) (i), candidates were expected to perform the basic *PUSH* and *POP* operations on a stack; (Module 1: Specific Objective 3). This question tested analysis and interpretation of the given operations. Generally, candidates were able to acquire total marks when they clearly illustrated the contents of the stack at various stages as the operations progressed. In a few cases candidates focused on the computation and neglected to show the contents of the stack. Marks were not awarded in those cases.

The responses to Part (b) (ii) were generally accurate. Most candidates were able to identify that there should be at least two elements in the stack in order for the *MULT* and *ADD* operations to work effectively. It should be noted that there were cases where candidates gave a vague response, for example, 'Stack should not be empty.' Marks were not awarded in these cases.

For Part (b) (iii), candidates were expected to analyse the above scenario and deduce that the contents of the stack would be reduced to one element and eventually the *ADD* operation will attempt to *POP* from an empty stack.

For Part (c) (i), the majority of candidates obtained four out of five marks because they were able to accurately identify that the *ENQUEUE* operation adds an element to the rear/tail of a queue and the *DEQUEUE* operation removes/retrieves an element from the front of the queue.

Candidates performed poorly on Part (c) (ii). The majority of their algorithms were incomplete and incoherent. The *DEQUEUE* function was not implemented properly and the iterative construct and its condition were in most cases incorrect or omitted. Generally, candidates were expected to:

- Initialize a ‘count’ variable
- Implement an iterative construct which will terminate when the queue is empty (‘Dequeue’ the queue correctly and Increment count)
- Return/Display count

Question 2

Part (a) tested candidates’ ability to explain how two numbers, 5 and 15, would be inserted into a singly linked list and then to represent the final list in the form of a diagram. Most candidates were able to give some level of explanation, however many were somewhat vague. In most cases they demonstrated the concept of a link between nodes but often were not clear on the sequence of steps involved in inserting items into the linked list. Candidates needed to clearly indicate in their explanation that a head was present and linked to either a node or null. Additionally, candidates needed to indicate that a new node must be created first then the number is stored in it and not merely “insert the number into the list”.

About 80 per cent of the candidates were able to produce reasonable diagrams to represent the nodes, however many did not correctly represent key components. For example, some candidates omitted arrow heads, while others had the head pointer arrow in the wrong direction. In some cases, the head pointer was drawn to look like a node; candidates needed to demonstrate some distinction between them. In other cases the notation used for the node was incorrect in that it did not clearly show the two parts, that is, the data and the address fields.

Part (b) tested candidates’ ability to declare an integer array, read values into it and then search for the presence of a particular value.

The majority of candidates were able to correctly declare an array. Candidates almost always indicated the correct data type as well as the array size. However, some candidates used the wrong name for the array although a specific one was given in the question.

In terms of storing items in the array which was the second requirement of this part of the question, most candidates demonstrated the need for a loop in order to perform this task efficiently. They were also able to produce the correct or near correct code to accept values and assign them to the array locations. However, many candidates omitted one or two key components such as the ampersand or the index. In some cases they also used the wrong format specifier instead of *%d*.

In the final section of Part (b), most candidates were able to produce C code to accept the search key. Some of them were able to write reasonable code to conduct the search, however quite a few used the wrong upper limit for the loop. They could have used < 100 since the C programming language starts to index an array from 0 causing index value 99 to represent the 100^{th} location.

For the comparison many used incorrect notation for ‘equal to’. Candidates should note that two equal signs ‘==’ are used for this purpose while a single ‘=’ is used for assignment.

For the last statement which should state ‘Key Not Found’, many candidates placed it within the loop so that the error message would be printed repeatedly. Other candidates did not have a statement to allow the program to break out of the loop after the key was found. This meant that the loop would continue processing unnecessarily after the key was found.

Section B – Software Engineering

Question 3

In general, almost all candidates attempted this question.

For Part (a), many candidates incorrectly wrote that a deliverable was the end product of the Systems Development Life Cycle (SDLC) to be ‘delivered’ to the customers. The majority of candidates did however identify a deliverable as some form of documentation or tangible/intangible result of each phase of the SDLC. Candidates lost marks for not indicating that a deliverable needed to be signed off and used as input for subsequent phases.

In Part (b), candidates responded well to the reasons why an information system may need to be changed. Candidates easily identified issues that would cause a system to change including change in requirements, inefficiency and outdated systems. Yet candidates could have done better by suggesting how a new system with modern updated technology would alleviate these issues.

In Part (c), while some candidates were able to correctly identify the factors: operational, technical, economic and legal, in some cases candidates’ responses lacked proper explanation of the terms and often candidates would confuse operational feasibility with technical feasibility.

For Part (d), the ERD was done quite well by most candidates. It should be noted that many candidates who should have received 12 marks lost marks for simple errors including:

- Plural entities (ending with ‘s’)
- Not indicating the primary key by underlining the attribute
- Incorrect cardinalities including the notation

Question 4

This question generally tested candidates’ knowledge of the design of input forms and of data flow diagrams (DFDs). Part (a) (i) was generally well done. However, some candidates provided one example only of data in the required formats.

Part (a) (ii) asked candidates to justify their selection of one of four input options. However, many candidates did not match the input option to a specific data item. Others misinterpreted the question and recommended different types of user interfaces for instance command-line or menu based.

Part (a) (iii) was poorly done. Most candidates incorrectly described range verification checks and did not relate this part of the question to the previous parts.

Part (b) tested candidates’ ability to create a data flow diagram based on a given scenario. Most candidates provided satisfactory responses, however many candidates incorrectly labelled the processes as departments; drew context diagrams which included each department listed as a source/external entity; and included departments along with their processes as separate symbols.

Candidates also lost marks for the data flow in the diagram because they incorrectly used verbs to describe the flow of data. Additionally, marks were not awarded when candidates used incorrect symbols for the four data flow diagramming objects (source/sink, data store, data flow, process).

Section C – Operating Systems and Computer Networks

Question 5

This question tested candidates’ knowledge of network devices, including: router, switch and modem along with their roles; hybrid network topologies, fibre optic cables, the IEEE802.11b network, VOIP, and GPRS.

Part (a) (i) dealt with setting up a small network using the following network devices: modem, router, and switch. Most candidates attempted this question, however, even though the devices were shown, some of them did not connect the devices correctly and critical ports were not shown.

Part (a) (ii) tested candidates’ knowledge of the role of the modem, switch and router. Most candidates demonstrated an understanding of the role of the modem being analog to digital conversion and vice versa. However, many candidates failed to adequately state the role of the router and switch.

Part (b) tested candidates' knowledge of the hybrid network topology. This question was well done. Almost all candidates were able to provide a correct definition of a hybrid network topology.

Part (c) tested candidates' knowledge of the advantages and disadvantages of fibre optic cables. This part was poorly done since candidates failed to adequately describe the advantages and disadvantages.

Part (d) was poorly done since many candidates failed to clearly explain how data is communicated in an IEEE802.11b network. Many of them did not demonstrate understanding of the use of wireless access points in this kind of network.

Part (e) required candidates to explain why the quality of voice over IP (VOIP) communication might be different from telephone communication service as offered by a telephone company. Most candidates demonstrated understanding that VOIP uses the Internet but failed to discuss the deterioration of the quality of the call where there are Internet problems or data packet delays.

Part (f) tested candidates' knowledge of the main purpose of GPRS. This question was fairly well done. Many of the candidates were able to link the term *GPRS* with the transmission of internet packets through a cellular network.

<u>Question 6</u>

This question tested candidates' knowledge of operating systems, security of data and device management. In addition it tested their knowledge of

- process management and in particular, process states
- interrupts and their application to scenarios
- scheduling processes with emphasis on pre-emptive scheduling.

For Part (a), most candidates were able to correctly identify that batch systems compiled jobs in groups. However, they were unable to specify that these jobs would be processed at a later time. The majority of candidates was also able to say that multi-user systems allow the use of many users at the same time but failed to identify their processing power.

For Part (b), most candidates recognized that a process may be blocked but were unable to define *a blocked process as a process that is paused, pending an event*. Candidates also failed to specify that *a running process is one where instructions were being executed by the CPU*. This part of the question was poorly answered by most candidates. Many candidates understood that the game would be halted to process the interrupt but most candidates did not identify the type of interrupt. Candidates also failed to recognize that a context switch occurs, that is, saving the state of the running process in registers for later resumption, and that the game process resumes after the processing of the interrupt.

Part (c) was poorly answered. While some candidates recognized that pre-emptive scheduling meant that the scheduler had the power to pre-empt the tasks, they did not specify that the processor would resume the previously running tasks.

In Part (d), the majority of candidates was able to correctly identify three ways of protecting data. However, some candidates were not able to sufficiently discuss their methods in order to gain full marks.

In Part (d), candidates were able to identify the type of software being used as device drivers. However, most candidates did not give the definition of a device driver as *the set of instructions used* *to communicate with the device. They also neglected to mention that each device driver uses the same protocol to communicate with the OS.*

Paper 03 – School-Based Assessment (SBA)

Requirements

Each student was expected to choose a problem for which a software solution exists and then develop the software using software engineering techniques. In particular, the student was expected to demonstrate appropriate choice of the tools and techniques used in the analysis of the software to be developed. They were then expected to design, code, and test their software using appropriate techniques.

General Comments

- Students should follow the order laid out in the criteria for marking when arranging the sections of the SBAs.
- Teachers should avoid using red ink pens to correct SBAs.
- Teachers should ensure that each SBA is clearly labelled with the student's name and centre number.

Marking Criteria

Specification of Requirements

<u>Definition of Problem</u>

Students were required to give a complete and accurate description of the problem. Most students handled this section fairly well but there were a few students still unclear as to what to include in their definition of the problem. A brief description of the context in which the problem occurred is required but details about actual problems staff/clients face and proposed steps to correct such problems must be emphasized.

<u>Techniques of Analysis</u>

Students were required to identify techniques of data collection and analyse the data collected.

- Most students were able to state the various techniques of data collection; however, some students were not able to justify their selections. Several students misinterpreted justification to mean definition. This is incorrect. Students should clearly explain why the chosen technique was used as it relates to the business/company/institution and not regurgitate advantages of the technique.
- Proof of analysis should also be given. All proof must be included directly after the analysis not in the appendix. For example, sample questions from the questionnaires and/or interviews should be included. It is understood that it may be difficult to include proof of observation; hence, if this technique is used, students should clearly describe what was observed.

<u>Use of Data Flow Diagrams and E-R Diagrams</u>

This section of the SBA was poorly done by most students.

Context Level Diagram

Students were required to give a complete and accurate diagram of all relevant entities and data flows.

- In some instances, incorrect symbols were used for entities and processes.
- Some data flows were not labelled.
- Data stores/files were incorrectly included in the Context Level Diagram.

Level 1 Diagram

Students were required to give a complete and accurate diagram with all relevant processes, data flows and major data stores.

- Most students are unaware that the Level 1 diagram is an expansion of the context level diagram. Hence, new external entities were created for the Level 1 diagram, as well as new data flows were created for previously used entities.
- Students had links between
  - data stores and entities
  - data stores and data stores
  - entities and entities

ER Diagram

Students were required to give a complete and accurate diagram of all relevant entities and relationships.

- Most students did not use the correct symbol for a relationship (a diamond).
- Some students did not include attributes for the entities.

<u>Functional and Non-Functional Requirements</u>

Students were required to give a complete and accurate description of all requirements.

- For the functional requirements, students did not clearly state what the system is expected to do; instead, they stated what the user will be doing. An example of a good functional requirement is *the system will be able to delete a patient's record.*
- For the non-functional requirements, students did not state the limitations of the system. An example of a good non-functional requirement is *the system can only store 1000 patient records*.

<u>Design Specification</u>

Students were required to give a complete and accurate system structuring diagram containing all processes and a description of the user interface, report design, algorithm design and appropriate data structures. Most students handled this section fairly well. However, the following should be noted:

- Students should not just include screens shots of interface and report design but also a justification in order to gain maximum marks.
- Narratives will not be accepted as an algorithm. Students are reminded to submit pseudocodes.

<u>Coding and Testing</u>

Students were required to produce a complete and accurate C program solution for the problem stated in the Definition of Problem section. Most students produced programs that achieved good functionality. Students should be aware that:

- All code must be written in procedural C. No other programming language will be accepted. Code must also be printed from the compiler, NOT transferred to a word processor before printing.

- Soft copies will not be marked.
- Most students did not include enough screen shots (and in some cases none at all) to support their functioning of the code.
- Some codes presented did not match the screen shots given. In such cases students were not awarded any marks.
- Test plans should be written in a tabular format and it should include normal, extreme, erroneous and incompatible data.
- All input data must be tested.
- Test results without related test plans will not be awarded any marks.

Communication and Presentation

This section of the SBA is often overlooked by students. Teachers and students are urged to pay close attention to their use of grammar and the overall presentation of the SBA.

Recommendation

Teachers should ensure that students are fully prepared for the examinations in both units. Poor performance in some modules of the syllabus indicates that more time needs to be allocated to these areas.

C A R I B B E A N   E X A M I N A T I O N S   C O U N C I L

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION® MAY/JUNE 2015

COMPUTER SCIENCE

GENERAL COMMENTS

This was the seventh year in which the revised syllabus was examined. There were three examination papers in each of Units 1 and 2, namely, Paper 01, Paper 02 and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the School-Based Assessment, was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of multiple-choice questions that were designed to test candidates' breadth of coverage of the syllabus. Paper 02 consisted of essay-type questions that were designed to test candidates' depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade are 30 per cent, 50 per cent, and 20 per cent, respectively.

Approximately 94 per cent of the candidates obtained Grades I─V in Unit 1 and approximately 94 per cent of the candidates in Unit 2 obtained Grades I─V. Overall, there is still need for improvement in the quality of responses for the programming questions in both units.

DETAILED COMMENTS

UNIT 1 – FUNDAMENTALS OF COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of approximately 64.3 out of 90, standard deviation of 13.9 and scores ranging from 24 to 90.

Paper 02 – Essay Questions

Section A – Computer Architecture and Organization

<u>Question 1</u>

Part (a) (i) required candidates to write down the truth table for a given security alarm system scenario. Most candidates attempted this part but many failed to use three inputs, using only two instead. Most candidates also failed to indicate clearly what labels represented, that is, switch, door or window. While most candidates produced a table with bit patterns, a few used phrases in their table, which were unacceptable.

Part (a) (ii) required candidates to use primary logic gates to design and draw circuit associated with (a) (i). Most candidates attempted this part but complicated their diagrams by using several gates, including NOT gates. Only two gates an OR and an AND gate were required. Some accurately produced diagrams were not labelled resulting in the loss of marks. In some instances, candidates' representation of the OR gate was unclear, closely resembling the NOT gate.

Part (b) required candidates to explain the term 'demultiplexer'. Most candidates attempted this part but some chose to simply say that it was the opposite of a multiplexor. Others gave long definitions of multiplexors and did not focus on the definition of the demultiplexer. In many cases, candidates incorrectly stated that it accepts one output signal to produce several input signals.

Part (c) required candidates to draw a clearly labelled block diagram of a 3 to 8 line decoder. Most candidates attempted this part successfully with some losing marks only for inadequate labelling. It was expected that all input and output lines, as well as the diagram itself be properly labelled.
Part (d) required candidates to determine the decimal equivalent of the hexadecimal number 1A53. Most candidates attempted this part, but the quality of the responses was generally unsatisfactory. While

most of the candidates were able to identify the base as 16, many could not identify the value of A as 10 and some did not apply the weighting of the first digit correctly.
Part (e) required candidates to find the 8 bit two's complement of -21. Most candidates attempted this part, but the quality of responses was generally unsatisfactory. While most candidates were able to correctly convert the number to binary, many were unable to go on to find the correct two's complement.
Part (f) required candidates to calculate the decimal equivalent of a given floating point representation.

Most candidates attempted this part. While most candidates were able to identify the exponent correctly, some were unable to determine the correct position of the decimal point by moving the point three spaces to the right, causing them to get an incorrect answer. Only a few candidates failed to recognize the number as being positive. Also, many candidates did not clearly separate and label the sign, exponent and mantissa components of the given binary representation.

Question 2

This question was attempted by the majority of candidates however the majority of them obtained between 11 and 15 marks with only about 5 per cent earning between 20 and 25 marks.

Part (a) tested candidates' ability to define the CPU and describe TWO of its main functions. Almost all candidates omitted to clearly define the CPU as hardware though they noted that it carries out the instructions of a computer programme. Candidates were able to identify the two major components of the CPU as the ALU and CU but the descriptions of the functions of each were often incomplete.

Part (b) (i) required candidates to state the purpose of the fetch operation while Part (b) (ii) required candidates to state the purpose of decode and execute operations. Candidates were generally correct in explaining the purpose of each operation, clearly stating that:

- The fetch operation is used to get data and instructions from main memory
- The decode operation determines what the given instruction means,
- The execute operation performs the instructions.

Part (b) (iii) required candidates to define the term *instruction set*. Part (b) (iii) was poorly answered as candidates failed to define the instruction set as a specific set of commands which the CPU is designed to understand.

Part (c) (i) required candidates to differentiate between ROM and RAM. The majority of candidates was able to identify two differences between ROM and RAM but some candidates confused the two and did not identify RAM as volatile and ROM as non-volatile.

Part (c) (ii) tested candidates' knowledge of different types of computer memory. Many candidates were unable to distinguish between cache and RAM and, hence, inadequately discussed the advantages and disadvantages of each option. Whilst many candidates indicated as an advantage, that for Option 1 having more cache would increase the speed of the computer as an advantage they did not identify being expensive as a disadvantage. For Option 2, many candidates did not recognize multitasking or cheaper costs as advantages of RAM but stated slower processing speeds as a disadvantage of having less cache.

Part (d) tested candidates' knowledge of the purpose and functions of registers found in the CPU. The majority of candidates correctly stated that registers hold all the data currently being handled by the CPU. However, about 70% of candidates were unable to correctly identify and describe registers such as the instruction register, program counter, memory address register and memory data registers.

Section B – Problem Solving with Computers

Question 3

Part (a) required candidates to explain what happens during the implementation and review phase of the problem-solving process. This question was poorly done by many candidates as they failed to correctly describe each phase. The, implementation phase was generally incorrectly described as identifying and selecting possible solutions. Reviewing was generally incorrectly described as the phase which involves the testing of the solutions. Much greater focus is needed in understanding and explaining the main activities in each stage of the problem-solving process.

Part (b) presented candidates with two algorithms and required them to identify which illustrated bounded and unbounded iteration. Candidates were also required to explain how the loops in the two algorithms would terminate. The majority of candidates was able to correctly identify the algorithm which were bounded and unbounded. However, many candidates were unable to correctly explain how the loops were terminated. For example, candidates incorrectly explained that the 'for' loop terminated on the 20th repetition. The 'for' loop terminates after the limit has been reached and on the final repetition when the limit is checked and is found to be outside of the range. The majority of the candidates correctly described how the 'While' loop was terminated. More focus in general must be given to understanding how loops are implemented.

Part (c) required that candidates identify and correct errors in a given algorithm. This question was generally well done by most candidates. However, greater focus is needed in writing the correct statements when errors are identified.

In Part (d), the candidates were required to write an algorithm which used iteration to find the sum of all multiples of 8 and all multiples of 11 between two positive integers. Candidates were required to sum the multiples of 8 and 11 within a given range. Many candidates were able to write statements to control the lower and upper limits of the range, as well as to determine the multiples of 8 and 11. However, more focus must be given to understanding the proper use of 'DIV' and division '/ '. These statements were used incorrectly on many occasions in an effort to determine if a given value is a multiple of 8 or 11.

Question 4

Part (a) tested candidates' ability to trace through the execution of a given algorithm and draw the output exactly as it would be generated by the algorithm. In particular, it tested candidates' ability to differentiate between functions which continue output on the current line and those which begin output on a new line. A key skill tested was the ability to traverse a loop accurately and determine the exit values of loop variables.

- This question was attempted by most candidates, however, many candidates failed to attain more than 50 per cent of the marks.
- Candidates failed to interpret the function of the output functions accurately even though they were clearly defined in the question paper.
- Many candidates iterated the loops too many times producing too many lines of output for each loop.
- Many candidates misinterpreted the end of the 'for' statement and included the next output statement as a loop statement even though indentation was used.
    - Inaccuracies produced were clearly linked to candidates' inability to calculate loop variables correctly.

Part (b) required candidates to construct a flowchart to represent a given algorithm. It tested candidates' ability to represent different control structures using flowcharts. Many candidates answered this question well gaining full marks. However, some candidates failed to demonstrate credible flowcharting skills. For example they:

- failed to accurately and consistently use the basic flowchart symbols
- included the use of the preparation symbol for initialization of variables and the display symbol for output
- used a circle as a start/stop or termination symbol
- neglected to use connection symbols completely
- often used the decision symbol without either the two flow lines proceeding from the symbol or did not label the flow lines y/n or t/f
- used flow lines without no arrows.
- incorrectly placed flow lines to show looping which also did not flow back to the decision in the structure.

Section C – Programming

Question 5

Part (a) tested candidates' knowledge of the lexical analysis process. Part (a)(i) tested knowledge of the purpose of the lexical analysis stage and Part (a)(ii) asked candidates to comment on the existence of error detection during the lexical analysis phase.

Part (a) most candidates did fairly well. Candidates were required to describe the process the lexical analysis goes through to convert a program from sources code to machine language. Most candidates failed to mention that during the lexical analysis process it converts a sequence of characters into a sequence of tokens which is read from left to right.

Part (b) was done well by most candidates. Those candidates who scored full marks were able to state what modulariation is and one of its benefits.

Part (c) was attempted by most candidates and was generally well done. Most candidates knew how to declare a file pointer, open a file in read mode and read from a file. Candidates who did well on this question were able to evaluate the grades using the appropriate selection control structure while accumulating each category.

Most candidates lost marks when they did not use an accumulator to track the number of grades for each category. Candidates also lost marks when they utilized a sentinel value instead of an end of file character in their looping condition.

Part (d) was poorly done. Most candidates who attempted the question used a string function to determine whether the string entered was a palindrome or not. Candidates lost marks when they made reference a single character instead of a 'string' inside their string function.

Candidates lost mark when they requested the length of the string entered instead of determining the length. Also, some candidates found it difficult to traverse the string to determine whether it is a palindrome.

Question 6

In general the question was poorly done with approximately 80% of the candidates scoring in the lowest range, that is, between 0 and 5 marks.

It was noted that while many candidates had a general sense of the logic, they were challenged with regard to the syntax of the language.

Part (a) required candidates to describe three types of programming languages. While most candidates attempted this section, most appear to be unclear in their understanding particularly of the scripting language. Most definitions were only partly correct.

Part (b) required candidates to trace through a program and record the output that would be produced. A large number of candidates were able to produce some part or all of the output. However, a significant number either could not do so or recorded the data in a trace table but did not display the results in the order they would appear on the screen.

Part (c) required candidates to write segments of code for an application that would allow a vendor to store data about some fruits that he sells. Each fruit was given an ID, cost price, selling price and quantity in stock. In writing the code candidates were expected to declare and use a C struct for the storage of data.

In Part (c) (i), while about 60 per cent of the candidates were able to declare the struct correctly many included items in the declaration that were not necessary or incorrect.

In Part (c) (ii) candidates had a fair understanding of how to declare the struct variables but often did not note the difference in syntax that was necessary for this section if they had used “typedef” in Part (c) (i).

For Part (c) (iii), candidates were required to assign values to the previously declared variable. Some did it through assignment while others prompted the user to enter from the keyboard as the question did not explicitly say how it should be done. Again, candidates had the logic in place but were challenged by the syntax of the language.

In Part (c) (iv), candidate were required to write code to exchange values that were previously loaded into the struct “mango” and “pine”. It was very clear that candidates could do the exchange if given an algorithm but many had difficulty doing so in the format that governs the C structure. This difficulty indicates a lack in their understanding of the language.

Finally in Part (c) (v), candidates were required to calculate and print the total profit to be made from the sales of the two fruits with preloaded data. Although candidates were quite creative in their solutions, many lost marks because their solution did not conform to C or were not logically sound.

In conclusion, candidates need to become more familiar with the language and in particular the use of structs for storing data.

Paper 03 – School-Based Assessment (SBA)

UNIT 1

General Requirements

Students were expected to choose a problem for which a software solution was appropriate; develop and present the solution in a logical way using correct grammar and appropriate jargon at all times by doing the following:

- Creating an algorithm for the solution using modules, sequencing, selection, assignments, and iteration (bounded and unbounded).
- Representing their algorithms using narrative format and also as either a flowchart or pseudo-code.
- Implementing the algorithm in C using various data structures such as arrays, struct, strings and files, with no less than five functions as independent units.
- Ensuring that the source code produced matched the algorithm.
- Createinga test plan with exhaustive data set, testing and producing test results with normal inputs giving correct results, extreme input giving correct results and erroneous input giving appropriate error message.

The following specific aspects of the project were assessed:

(a) Definition of problem
(b) Narrative and flowcharts or pseudocode
(c) Coding of program
(d) Testing and presentation
(e) Communication of information

For each component, the aim was to find the level of achievement reached by the student. It is recommended that the assessment criteria be available to student at all times.

DETAILED COMMENTS

<u>Definition of Problem</u>

Many students were able to accurately describe the problem complete with the description of the current system and giving background to the problem, the issue or problem, recommend solutions and examples of what takes place when the problems arise. However some students projects only included partial definitions of activities of the proposed system that they intended to create without complete descriptions of the current system and the problems which are to be solved using software. In addition, some problems that were identified would not be solved by using a computer program. For example, filling out of a form is a time consuming and error prone process for both a paper and computer-based form, however, there are advantage of using computer-based forms that were not identified.

<u>Narrative and Flowcharts or Pseudocode</u>

Narratives were fairly well written in most samples. However some samples:

- Provide only a partially correct description of the algorithm as a solution
- Provide description of an intended module or unit instead of describing the step-by-step procedure that would solve the problem as required by an algorithm

Pseudocode algorithms were generally well done. Samples which included flowcharts in some instances incorrectly used some symbols. For example, system flowchart symbols were used and in others cases, diagrams were poorly presented, had symbols with no labels such as (yes/no for the decision symbol), no connector symbols to connect diagrams to the next page and, therefore, the charts were difficult to follow. Most students produced samples using pseudocode algorithms, some samples were clearly modified copies of the programming source code, with a few almost identical copied of the source code. It is therefore recommended that in the future, students be encouraged to develop their algorithms independent of the programming code.

Coding of Program

Students' projects demonstrate that they were comfortable with procedural C programming language. This was quite evident by the response of the various projects. Most programs were logically written and properly decomposed. Nevertheless, some candidates sample did not

- comprise functions as independent units and some teachers still awarded full marks
- provide sufficient data structure in the program
- demonstrate appropriate use of the concept of structured programming using evidence of sequence and selection but no loops
- use appropriate data structure (struct, files and arrays). Only a few samples made use of all or most of the key data structures.

Some students did not print their project from the C compiler, instead choosing to make print screen copies of their projects. Students are therefore reminded to print their source code directly from the C compiler as transferring the code to a word processor often changes the code. For example, it can adjust the spacing and strange symbols to occur in the code, making it difficult to read. This may also cause students losing marks for appropriate programming style and documentation. In addition, marks were lost because some students did not include adequate comments at key areas of their source code and used a poor indentation style that did not make the code easier to read. However this section was done well by most students.

Evidence that Code Matches Algorithm:

Most students were able to obtained full marks on this section for this was well done.

Evidence of File Manipulation:

Students were required to provide code that included a file and show evidence of file manipulation, that is, open file, write to file, read from file, append file and close file. Most students demonstrated evidence of this, however some students were awarded marks by teachers even though there was no evidence of file manipulation and in some cases the use of file was none existent or was declared but not used, non-existent but teachers awarded full marks. Projects that showed screenshots of the file data being added, edited and deleted were awarded full marks.

Testing and Presentation:

Students were required to prepare a test plan with an exhaustive data set (test data) to test their program. They were also expected to use the test data set to produce test results with normal input giving correct results, extreme input giving correct results and erroneous (abnormal) input giving appropriate error messages. The majority of samples provided this year did not have a suitable range of test data, but a few candidates:

- Had a test plan and screenshots while others provide screenshots without any test plan
- Did not include all testing criteria (normal, extreme and erroneous) in their test plan
- Had test results but did not have a clear test plan
- Had no test results but candidate were awarded marks by the teacher
- Had test results which did not include actual screenshots of the working program and testing was not done using the test data outlined in the test plan
- Were awarded marks for a test plan, when they had no plan
- Therefore teachers may have not understood the marking criteria or did not understand what a test plan should be
- Only tested their menus with the appropriate test data, but all input must be tested in the same way

Generally, this year's IA projects were well presented. Twenty-five per cent of the candidates gained marks between 51 to 60, twenty eight per cent had marks between 41 to 50, twenty two per cent had marks between 31 to 40 and only twenty five made less than 31.That is about 75 per cent of the candidates made a mark of over 30, this demonstrates that most IA were satisfactorily done.

Recommendation

Teachers need to ensure that the students use the headings outlined in the criteria for marking the SBA Project in the syllabus, as well as follow the order in which these headings occur on Form CSCI 1−3. They should also ensure that students, check that the numbering in their table of contents numbering correspond to the numbering in the document and candidates provide information in a logical way using correct grammar and appropriate jargon at all times in the presentation of their projects.

UNIT 2 – FURTHER TOPICS IN COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of 64.4 out of 90, standard deviation of 12.3 and scores ranging from 21 to 88.

Paper 02 – Essay Questions

Section A– Data Structures

<u>Question 1</u>

Part (a) required candidates to describe the concept of ADTs (Abstract Data Types). Most candidates interpreted the 'D' to mean dynamic and spoke about changes in the size of the stack or queue during program execution. Some candidates were able to give basic explanations but could not expand further.

Part (b) required candidates to describe the differences between stacks and queues and give real world examples. This question was very well done and most candidates were able to identify and describe the differences; however, a few candidates struggled to give correct real world examples of stacks. Most candidates got full marks for this question.

Part (c) (i) required candidates to write C code for a stack's PUSH function. The logic suggest that the function must first check that the stock is full then increment the top position before storing the value. These steps were often confused. Candidates struggled to write correct function prototypes (arguments and return type) and write valid C code. Some candidates assumed the existence of a separate function (isFull) when the question specified not to use predefined functions.

Part (c) (ii) required candidates to write C code for a queue's ENQUEUE function. Most candidates did not consider a circular queue. The logic suggest that the function must first check that the queue is full then increment the tail position before storing the value. These steps were often confused. Candidates struggled to write correct function prototypes (arguments and return type) and valid C code. Some candidates assumed the existence of a separate function (isFull) which the question specified not to do. Some candidates gave the same function as Part (c) (i) which added to the top and not the rear/tail.

Part (d) required candidates to demonstrate how a stack's POP operation works on a stack containing four letters. The majority of candidates was able to answer this question completely. Some candidates began popping (removing letters) the stack from the incorrect end. Some candidates did not show the final contents of the stack after all the POP operations were completed.

<u>Question 2</u>

Part (a) tested candidates' ability to sort a given array using the selection sort algorithm, but only displaying the first three passes. Many candidates did not perform the swap correctly by displaying two integers exchanging places, instead many showed one move of an integer which did not earn them the full two marks. Some candidates did not realize that on the 3 pass, the position of the integers remained in place and no swapping took place. Some candidates misunderstood what was required from the question and chose instead to draw 3 instances of the array on the 4, 5 and 6 pass, thus earning themselves no marks. Some candidates did not comprehend the selection sort algorithm at all, gave erroneous responses and so scored no marks. About 6 per cent of the candidates earned full marks on this part.

Part (b) (i) tested candidates' ability to draw a linked list containing three nodes. In Part (b) (i), the majority of responses earned the candidates one or more marks, but many did not correctly indicate nodes as having two parts: data and pointer − and that between two nodes an arrow (not just a line) pointed to the other node. Some candidates' indicated two arrows flowing in opposite directions between the nodes (a doubly linked list), which was acceptable and earned them the mark. The head is a pointer (and not a node) and was not indicated as such in many candidates' responses. Some candidates incorrectly indicated the end node (node containing integer 5) as pointing to a null node. Acceptable responses indicating the end node were the word NULL, the earth or ground symbol, an X, 0 or Ø.

In Part (b) (ii) candidates were asked to explain how the first integer from the linked list could be deleted. This was not well done; in that about 90 per cent of candidates' responses did not earn them full marks. Candidates needed to search for the node (node 3), set a temporary pointer to point to the head node (node 3), set the original head to point to 4, then free the memory allocated to 3.

Part (c) had two parts and required that candidates write code to obtain a value **(target)** entered by the user, and to perform a search, given a sorted array (**numbers)**. Most candidates scored full marks for writing their code, correctly using scanf, and &target. In Part (c) (ii), however, about 10% of the responses earned full marks, many candidates did not understanding the binary search concept. Teachers should be reminded that C is the required language for CAPE Computer Science, and that other languages including pseudocode are unacceptable. Candidates should also note that where the question posed refers to a specific variable, the use of these variables is mandatory in the required solution.

Section B – Software Engineering

Question 3

For Part (a), candidates were required to explain the rapid prototyping approach (RAD to systems development. The majority of candidates was able to describe RAD as a model/simulation of a system. Candidates lost marks for not specifying that it was a quick fabrication of what the real system might be like.

In Part (b), candidates were required to describe the main challenge which RAD seeks to overcome. While some candidates were able to identify that the RAD approach helped clarify customer requirements, other candidates responded that the process was time consuming and a waste of resources.

For Part (c), candidates were required to give FOUR reasons why it is believed that the SDLC may be initiated and completed during the maintenance. Some candidates were able to correctly identify the four major points that is, analysis, design, coding and testing which are conducted during the maintenance phase of SDLC. However candidates' responses lacked proper explanation of how each point could be integrated in the maintenance cycle. For example, when maintenance is conducted the analysis phase is initiated as new requirements must be gathered; feasibility tests and scheduling and so on must be carried out.

For Part (d), candidates were required to draw a Level−0 DFD to describe the flow of data in a given scenario. The Level−0 diagram was very poorly drawn as the rules for constructing these diagrams were often ignored. Many candidates:

- Confused Level−0 with Context diagrams
- Used inconsistent and incorrect symbols to represent sources, data stores and processes
- Used verbs for data flows
- Connected sources directly to each other
- Drew processes with data in-flows only

Question 4

Part (a) tested candidates' knowledge of the significance of data flow diagrams (DFDs) in the Analysis and design phases of the SDLC. Most candidates were unable to distinguish between the purposes of the DFDs in these phases, incorrectly stating that DFDs in the analysis phase depict how the system 'should' operate. However, some candidates were able to explain that DFDs can be used in comparative analysis which can in turn lead to business process re-engineering.

Part (b) required that candidates state three benefits of the object−oriented approach to software design such as reusability, maintainability and scalability. Many candidates did not answer this question well; most candidates provided answers that were too vague, for example, objects are easier to use and code.

In Part (c) candidates were asked to describe two (2) of the major deliverables from the design phase of the SDLC. Many candidates either omitted this part of the question or incorrectly identified various components, tools and techniques (including HIPO charts, DFDs, algorithms, interface design and prototypes) as major deliverables. Many responses included the feasibility study or report which was incorrect, as this is done in the analysis phase. Few candidates were able to correctly identify and describe the system architecture and design specification as major deliverables resulting from the design phase.

Part (d) required that candidates draw an ERD based on a given scenario. Candidates were able to gain most of their marks for this subpart as opposed to Parts (a), (b) or (c). Though most candidates showed major improvements when compared with last year's ERD diagram question, additional emphasis should be placed on the consistency of the symbology. Common errors included the following:

- Pluralization of individual entity names: EMPLOYEE not EMPLOYEES.
- Vague relationship descriptions: has, is

Section C – Operating Systems and Computer Networks

Question 5

This question tested candidates' knowledge of types of networks, network architectures, the OSI model, routers and GPRS.

Part (a) required that candidates describe two advantages of a client/server network over a peer to peer network. This question was poorly answered as many candidates were vague in their responses. Candidates confused the client/server network and the peer-to-peer network with network topologies. They also failed to indicate that the centralized nature of the client/server network allowed for easier management. While many candidates mentioned security as an advantage, few included that special software would be needed for better security control.

In Part (b) (i) candidates were asked to define the term *firewall*. Most candidates acknowledged that it was a hardware/software security measure but were unable to specify that it protects against external network threats.

In Part (b) (ii) candidates were asked to explain why firewall software needs to be updated frequently. This question was well answered with the majority of candidates correctly identifying that since new malware, viruses and threats are created daily via the Internet, firewall software must be updated.

Part (c) required that candidates state two differences between FDDI and Ethernet network architectures. Many opted not to answer this question and those who did failed to describe the architecture of FDDI and Ethernet, instead they described the speeds and physical cabling. Few candidates gave the correct answer in their response, that is, FDDI has a dual ring layout with the second ring acting as a back-up while Ethernet does not have a back-up configuration like the secondary ring of FDDI.

Part (d) dealt with the role of the network and transport layer of the OSI model. Some candidates failed to differentiate between the layers and did not include the role of the network layer in path determination and IP addressing. They also failed to discuss the transport layer's role in breaking up a file into parts and the reliability of packages in transport. Few candidates specified the particular protocol associated with each layer.

Part (e) was well answered by most candidates. This question required that candidates outline three benefits of wireless routers over wired routers. Few candidates confused the mobility of the devices on the network with the mobility of the router. However, most were able to correctly identify the portability of the devices, the inconvenience of installation, the compatibility issues with newer devices and the susceptibility of cables to damage as valid answers.

Part (f) (i) dealt with the purpose of GPRS. Most candidates were able to identify that GPRS was used in the transmission of data but failed to elaborate that it is a wireless technology used for data transmission through a cellular network. A few candidates mistook GPRS for GPS.

In Part (f) (ii) required candidates to name and describe two applications of GPRS. Most candidates were able to gain marks by correctly identifying applications such as point-of-sale terminals, MMS, tracking and navigation devices, web browsers and wireless electricity meters as solutions. However, they failed to describe fully how they operate in order to gain full marks. For example, POS devices are portable devices which allow customers to make payments without the use of a phone line. This device uses a spectrum that one of the carriers has licensed and a fee has to be paid.

<u>Question 6</u>

This question tested candidates' knowledge of device management, file security and user interfaces. In addition it tested their knowledge of:

- Process management and in particular components in a process control block (PCB)
- Scheduling algorithms with emphasis on round-robin scheduling algorithms

Part (a) required that candidates explain the role of a device driver in an operating system. Most candidates recognized that a driver was software that allowed the operating system to communicate with a device but failed to note that drivers provided the operating system with instructions on how to control the device.

Part (b) tested candidates' knowledge of spooling. This question was generally well done as many candidates noted this was where documents where loaded into temporary storage location pending an I/O device availability.

Part (c) was also generally well done as many candidates were able to outline TWO advantages of a menu interface over a command interface. However, candidates frequently made mention of 'lines of code' or a 'machine language' instead of *commands* when referencing typing instructions in a command interface.

Part (d) asked candidates to describe TWO components of a process control block (PCB) in an operating system. Many of them supplied suitable components that is, process ID, process state, and registers. However, many candidates correctly identified the program counter by itself as a component.

In Part (e) candidates correctly mentioned encryption, access control lists and firewalls as suitable ways in which files can be protected from unauthorized access. However, many failed to explain how a firewall prevents access to files.

Candidates also made mentioned of activity logs and file compression, methods which by themselves would not protect files from unauthorised access to a file.

Part (f) was generally well done, with the majority of candidates recognizing that each process is given a certain amount of CPU time (time slice or quantum). They were also able to provide adequate details of how the scheduling algorithm functioned.

In Part (g), candidates were asked to explain how an interrupt is handled by the process in the computer. Many candidates provided generally suitable answers. However, quite a few of them instead described what an interrupt is or identified instances that would cause an interrupt in the system as opposed to focusing on what happens in the processor when an interrupt occurs. Namely, a context switch occurs or the running process is saved in registers for later resumption, after the processing of the interrupt.

Paper 03 – School-Based Assessment (SBA)

UNIT 2

Each student was expected to choose a problem for which a software solution exists and then develop the software using software engineering techniques. In particular, students were expected to demonstrate appropriate choice of the tools and techniques used in the analysis of the software to be developed. They were then expected to design, code, and test their software using appropriate techniques.

Definition of Problem:
Students were required to give a complete and accurate description of the problem. Most students handled this section fairly well but there are a few students who are still unclear of what to include in their definition of the problem. A brief description of the context in which the problem occurred is required but details about actual problems staff/clients face and proposed steps to correct such problems should be emphasized.

Techniques of Analysis:
Students were required to identify techniques of data collection and analyse the data collected.

Most students were able to state the various techniques of data collection; however, some students were not able to justify the reasoning for their selection. Several students misinterpreted justification to mean definition. This is incorrect. Students should clearly explain why the chosen technique was used as it relates to the business/company/institution and not regurgitate advantages of the technique.

Proof of analysis should also be given. All proof must be included directly after the analysis and not *in the appendix*. For example sample questions from the questionnaires and/or interviews should be included. It is understood that it may be difficult to include proof of observation; hence, if this technique is used, students should clearly describe what was observed.

Use of Data Flow Diagrams and *ER* Diagrams

This section of the SBA has improved significantly. Students were able to properly draw relevant diagrams that correctly mapped to their problem statement.

Context Level Diagram

Students were required to give a complete and accurate diagram of all relevant entities and data flows. The weaker students generally failed to label their data flows.

Level-1 Diagram

Students were required to give a complete and accurate diagram with all relevant processes, data flows and major data stores.

Most students are unaware that the level 1 diagram is an expansion of the context level diagram. Hence, new external entities were created for the level 1 diagram, as well as new data flows were created for previously used entities.

Students had links between (1) data stores and entities, (2) data stores and data stores, (3) entities and entities, which are *all* incorrect.

Some students' diagrams were too small, which made reading and marking difficult.

ER Diagram

Students were required to give a complete and accurate diagram of all relevant entities and relationships. Most students did not use the correct symbol for a relationship (a diamond). Some students did not include attributes for the entities.

Functional and Non-Functional Requirements:

Students were required to give a complete and accurate description of all requirements. For the functional requirements, students did not clearly state what the system is supposed to do but instead stated what the user will be doing. An example of a good functional requirement is the system will be able to delete a patient's record. For the non-functional requirements students did not state the limitations of the system. An example of a good non-functional requirement is the system can only store 1000 patient records.

Design Specification

Students were required to give a complete and accurate system structuring diagram containing all processes and a description of the user interface, report design, algorithm design and appropriate data structures. Most students handled this section fairly well.

Most students were unaware of how to correctly draw a system structuring diagram. Students should note that this diagram is similar to the HIPO chart as it shows the breakdown of the system into its submodules and the submodules into its further components.
Students should not just include screenshots of interface and report design but also a justification in order to gain maximum marks.

Narratives will not be accepted as an algorithm. Students are reminded to submit pseudocodes.

Coding and Testing

Students were required to produce a complete and accurate C program solution for the problem stated in the Definition of Problem stage. Most students produced programs that achieved good functionality. Only procedural C is accepted as the programming code. No other programming language will be accepted. Code must also be printed from the compiler, not transferred to a word processor before printing. Students should not include soft copies of their SBA. Soft copies will not be marked.

Most students did not include enough screenshots (and in some cases no screenshots) to support their functioning of the code. Some codes presented did not match the screenshots given. In such cases students were not awarded any marks.

Test plans should be written in a tabular format. It should include normal, extreme, erroneous and incompatible data. All input data must be tested. Test results without related test plans will not be awarded any marks.

Communication and Presentation

This section of the SBA is often over looked by students. Lectures/teachers and students are urged to pay close attention to the use of grammar and the overall presentation of the SBA.

Further Comments

i. Students should follow the order and the headings laid out in the criteria for marking when compiling the SBA.
ii. Students should avoid using watermarks when printing the SBA as it makes it difficult for the examiner to read and assess their work.
iii. Lecturers/teachers should avoid using red ink pens to correct the SBA.
iv. Lectures/teachers should ensure that each SBA is clearly labelled with the candidate's name and centre number.

Recommendations

Teachers should ensure that students are fully prepared for the examinations in both units. The poor performance in some modules of the syllabus indicates that more time needs to be these areas.

C A R I B B E A N E X A M I N A T I O N S C O U N C I L

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION®
MAY/JUNE 2016

COMPUTER SCIENCE

GENERAL COMMENTS

This was the eighth year in which the revised syllabus was examined. There were three examination papers each in Units 1 and 2; namely, Paper 01, Paper 02 and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by the Caribbean Examinations Council (CXC) while Paper 03, the School-Based Assessment, was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of 45 multiple-choice questions that were designed to test candidates' breadth of coverage of the syllabus. Paper 02 consisted of six compulsory essay-type questions that were designed to test candidates' depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade are 30 per cent, 50 per cent, and 20 per cent, respectively.

Approximately 91 per cent of the candidates obtained Grades I–V in Unit 1 and approximately 95 per cent of the candidates in Unit 2 obtained Grades I–V. Overall, there is still need for improvement in the quality of responses for the programming questions in both units.

DETAILED COMMENTS

UNIT 1 – FUNDAMENTALS OF COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of approximately 64.26 out of 90, standard deviation of 13.57 and scores ranging from 18 to 90.

Paper 02 – Essay Questions

Section A – Computer Architecture and Organization

Question 1

Part (a) required candidates to write down the truth table for a given combination of switches. Most candidates attempted this part of the question but many did not interpret the switches correctly and often noted the wrong combination of bit patterns. In many cases the columns representing the inputs were not correctly labelled.

Part (b) required candidates to write down the truth table for a given proposition. Most candidates attempted this part of the question but it was clear that many confused the operand 'AND' with 'OR' or did not understand the proposition at all.

Part (c) required candidates to define the term 'program counter'. Most candidates attempted this part of the question but were not always clear in stating that a program counter is a register that stores the address of the next instruction in memory. Some candidates confused this term with a loop counter and in these cases marks were not awarded.

Part (d) required candidates to add two 8-bit binary numbers stored in Registers 1 and 2 respectively. Most candidates were correct in adding the two numbers together but marks were lost in explaining what would be stored in Register 3. While the calculation generated a 9-bit binary number, Register 3 would store 8-bits only. As a consequence, an overflow error would occur and the overflow bit on the left would be truncated. Alternatively, candidates could indicate that nothing would be stored because of the overflow error.

Part (e) required candidates to determine the binary equivalent of the decimal number 13.25. Most candidates attempted this part of the question but lost marks for lack of working despite this being clearly requested in the question posed. Many candidates successfully converted the whole number part of the given decimal number but failed to correctly convert the fractional part. In some cases, the same conversion technique used for the whole number component was used to convert the fractional component thus yielding the wrong answer.

Part (f) required candidates to define the term 'multiplexor' and draw a block diagram to support the answer. Most candidates attempted this part of the question by defining a multiplexor as a device which selects one of several input signals and forwards the selected input into a single line. Though the supporting diagram was accurately presented and many candidates correctly illustrated 1 output channel, several candidates failed to correctly represent 2^n input channels.

Part (g) required candidates to explain why multiplexing was important. The question challenged candidates to apply their knowledge of multiplexing by providing a real-world example. In general this question was poorly answered. Central to the solution was the understanding that multiplexing makes better use of available bandwidth across a communication network. Specific scenarios included the provisioning of cable TV and telephones as well as CCTV cameras. A further explanation to illustrate the use of multiple inputs versus a single output was required to support the given scenario.

Question 2

Candidates were presented with a diagram in Part (a) and asked to provide examples of non-volatile and secondary memory. This question was generally well answered by all candidates, with various versions of ROM presented as examples of non-volatile memory, and hard disks, floppy disks, USB drives, CDs, DVDs and SSDs (Solid State Disks) presented as examples of secondary memory.

Part (b) described a scenario in a fictitious community where electricity was available but inconsistent. The details of a cell phone were provided and candidates were asked to explain the purpose and benefit of the using each of three devices in the scenario. It was therefore important to link the use of the devices to the given scenario. Most candidates attempted this question but many were not clear in stating the benefit of the device in the context of the scenario.

In Part b (i), candidates were expected to note that a surge protector would limit any unwanted voltage above a safe threshold, thus protecting the cell phone in the case of a power spike. In Part b (ii), candidates were expected to note that a voltage regulator automatically maintains a constant voltage and is a proactive device that provides clean electricity regardless of changes to its input voltage. In Part b (iii), candidates were expected to note that an uninterruptible power supply is a device that allows the cell phone to keep running for a short time when the primary power source is lost. The device could therefore be charged by the UPS for a limited time.

Part (c) required candidates to define the term 'clock speed' as it relates to the processor and give examples of CPU clock speeds using different units of measure. In general, most candidates attempted this question and stated that the clock speed is the speed at which the processor executes each instruction. Candidates were expected to give two distinct examples, such as 2.5 MHz and 1.8 GHz to demonstrate their knowledge of the different units. Candidates lost 1 mark where distinct units of measure were illustrated but no quantifying speed was given.

Part (d) required that candidates apply their knowledge of input/output devices by providing examples of devices that would be appropriate for a person who is visually impaired. Candidates were expected to identify the device and describe how it would be useful in this case. Most candidates attempted this question, with many scoring full marks. The notion of voice to text and text to voice were important to note as these points indicated a clear knowledge of the requirements of a visually impaired person.

In Part (e), candidates were asked to explain, with the use of examples, the main difference between a supercomputer and a mainframe computer. Most candidates attempted this question and gave reasonable examples of the use of these systems though many failed to highlight the main difference clearly. Candidates were expected to note that the main difference between the two systems was the type of problems each tries to solve. Candidates were able to achieve only partial credit when the cost difference between the two types of systems was emphasized. In general, candidates were expected to note that supercomputers are ideal for complex calculations such as weather forecasting/predictions while mainframe computers focus on problems which are limited by input/output and demand reliability.

Part (f) asked that candidates use the processor's instruction set to write a set of instructions to perform a mathematical calculation. This question was poorly answered as many candidates did not demonstrate a knowledge of appropriate instructions that could be used. Candidates were expected to use an appropriate mnemonic to represent instructions which included specific operands (registers) as is expected for calculations. For example,

```
LOAD R1, R2
MULTIPLY R1, R2
STORE R1
```

Here LOAD, MULTIPLY and STORE are the instructions used while R1 and R2 refer to registers.

Section B – Problem Solving with Computers

<u>Question 3</u>

The scores on this particular question were not as evenly distributed as those recorded for previous questions. That is, though a reasonable number of candidates scored between 15 and 25 inclusive, almost 10 per cent of the total number of candidates scored 0, while many did not attempt this question at all.

Part (a) required candidates to write an algorithm to read a given set of data related to library fines. The data comprised a customer ID, number of books outstanding and a category corresponding to the rate at which each fine was charged. Results required included the total outstanding fine for each customer, total amount of fines paid by category, total amount of fines, number of customers and customer id for the highest fine paid. Many candidates recognized the need for an unbounded looping structure in this case but at times confused the loop condition that should be used (customer ID < > 9999). Many candidates did not include the input statement before the start of the looping structure and often omitted the input at the end of the loop. IF statements within the looping structure were generally well used, as were cumulative variables. Many candidates did not recognize, however, that outputting the ID of the customer with the highest fine would require both the customer ID and customer fine to be stored in order for the correct comparison to be made.

Part (b) asked that candidates distinguish between 'bounded' and 'unbounded iteration'. Many candidates noted examples only and this was not sufficient to answer the question. In distinguishing between the two terms, it was important for candidates to note that though both terms referred to a looping structure with bounded iteration, the number of passes through a loop is known beforehand while with unbounded iteration the number of passes through the loop is not known beforehand.

Part (c) required that candidates write an algorithm to find the sum of all multiples of 7 between 21 and 210 inclusive. Many candidates recognized the need for a looping structure in this case and, though not required, demonstrated their knowledge of the modulus function and this was commendable. Several candidates did not initialise the counter variables used and therefore failed to earn the maximum marks on this part of the question. There were a few candidates who did not understand the caveat to include 21 and 210 in the summation and thus provided an incorrect loop condition.

<u>Question 4</u>

Part (a) presented an algorithm and required candidates to construct a flowchart to represent it. Candidates were assessed on appropriate signals, as well as the ability to accurately represent the logic. A few candidates satisfied all the assessment requirements and achieved the maximum score available for this part. Often, however, candidates made a few common mistakes, such as:

- Failing to recognize that two separate conditions are represented in the algorithm
- Failing to place correct labelling on decision outputs.

In addition, a few candidates had no arrows on flow lines and many candidates did not use arrows to properly represent the loop structure.

Part (b) required candidates to write an algorithm to perform a calculation that finds the cost based on an input variable and a condition that specifies whether one formula or another will be applied. Some of the errors observed were:

- Many candidates achieved full marks on this question.
- A high number of candidates represented one or more formulae inaccurately.
- A few candidates did not print the output.
- A few candidates did not recognize that a condition structure was to be applied.

Part (c) required that candidates describe four steps in the software development process. Many candidates were able to state four steps, but only a few followed through properly with a description of all the mentioned steps. Some candidates stated the steps, but did not provide a description.

Section C – Programming

<u>Question 5</u>

Part (a) described the requirements for a system that generates a random number and then allows a user to repeatedly guess the number until the guess is correct.

A few candidates scored fairly well. A high number of candidates, however, provided answers that contained minor syntax errors. Many candidates described syntactically incorrect loop conditions. Some of the candidates did not provide appropriate output and omitted the loop structure. A small number of candidates generated the random number in the wrong part of the program (for example, in the loop).

Part (b) (i) asked candidates to explain what a procedural language is. A high number of candidates correctly indicated step-by-step operation of a computer program; however, some candidates did not give a sufficient explanation.

Part (b) (ii) required candidates to describe object oriented programming. Some candidates scored full marks, but an inordinately high number of candidates did not give a satisfactory answer.

Part (c) required that candidates write a program that opened a formatted file, read the contents, and printed some summary information. Few candidates scored full marks on this question. Some candidates did not properly open the file. Some candidates did not initialize variables properly. Many candidates used an improper loop condition (for example, using the equality test to compare the contents of a string). Many candidates did not read the data properly from the file. Many candidates did not properly test for the exit condition for the program. A few candidates did not properly aggregate the summary data, and a few candidates did not print the output properly.

<u>Question 6</u>

Part (a) of the question required candidates to state three differences between applications for mobile devices and those developed for desktop computers. A large number of candidates were able to properly describe two differences, the most common ones being the design for smaller screens, the design with limited memory, and the differences in input devices. A few candidates misinterpreted the question and provided solutions that described differences in the devices, rather than applications for the devices.

Part (b) required candidates to write down the output of a program that was presented. Few candidates provided a complete answer to this question. Most candidates, however, were able to produce some correct values generated on the first line.

Part (c) required candidates to write code that defined and manipulated a structure that stored information on books.

Part (c) (i) required candidates to declare the structure, and a large number of candidates declared the structure properly. Some candidates used the typedef command correctly, which affects where the structure name “BookRec” is expected, and also affects manipulation in other sections. Some candidates who did not use the typedef command did not name the structure before specifying contents.

Part (c) (ii) required candidates to populate the structure. A large number of candidates completed this task properly, but some candidates did not reference the data in the structures correctly.

Part (c) (iii) required candidates to increase a numeric data item by a percentage. Many candidates answered this question properly. Some candidates did not reference the data item properly. Some candidates also did not use the correct formula when increasing the percentage.

Part (c) (iv) expected candidates to write a code that would swap the values in two records. Candidates who completed the task properly would have taken on two primary approaches, namely, using an auxiliary record and swapping the entire record, or using auxiliary data items to swap individual items within the record. Some candidates presented incorrect swap logic. Some candidates did not reference data correctly. Some candidates did not declare auxiliary data structures or variables properly.

Paper 03 – School-Based Assessment (SBA)

General Requirements

Candidates were expected to choose a problem for which a software solution was appropriate and develop and present the solution in a logical way, using correct grammar and appropriate jargon at all times. In accomplishing this they were required to do the following:

- Create an algorithm for the solution, using modules, sequencing, selection, assignments, and iteration (bounded and unbounded).
- Represent their algorithms, using narrative format and also as either a flow chart or pseudo-code.
- Implement the algorithm in C, using various data structures such as arrays, struct, strings and files, with no less than five functions as independent units.
- Ensure that the source code produced matched the algorithm.
- Create a test plan with exhaustive data set; test and produce test results with normal inputs giving correct results, extreme input giving correct results, and erroneous input giving appropriate error message.

The following were specific aspects of the project that were assessed:

(a) Definition of problem
(b) Narrative and flow charts or pseudocode
(c) Coding of program
(d) Testing and presentation
(e) Communication of information.

For each component, the aim was to find the level of achievement reached by the candidates. It was recommended that the assessment criteria be available to candidates at all times.

DETAILED COMMENTS

Definition of Problem

Many candidates were able to accurately describe the problem complete with the description of the current system, and give examples of what takes place when the problems arise. However, some candidates' projects only included partial definition of activities of the proposed system that they intended to create without completely describing the current system and the problems or issues which are to be solved using software.

Narrative and Flow Charts or Pseudocode

Narratives were fairly well written in most samples. However, some samples did not provide a completely accurate description of the algorithm. Proper algorithm structure was not used. For example, many algorithms submitted had no name. Some only provided descriptions of each intended module or unit instead of describing the step-by-step procedure that would solve the problem as required by an algorithm.

Pseudocode algorithms were generally well done. Samples which included flowcharts in some instances incorrectly used some symbols, for example, system flowchart symbols, and in other cases diagrams were poorly presented, had symbols with no labels (such as yes/no for the decision symbol) or no connector symbols to connect diagrams to the next page and therefore were difficult to follow.

In the many samples using pseudo-code algorithms, some were clearly modified copies of the programming source code, with a few being almost identical copies of the source code. Candidates should be encouraged to develop their algorithms independent of the programming code, using proper algorithm structure and design.

Coding of Program

Candidates' projects demonstrated that they were comfortable with procedural C programming language. Most programs were logically written and properly decomposed. Nevertheless, some candidates' samples did not comprise functions as independent units and, despite this, were still inappropriately awarded full marks by the teacher, while some projects had few data structures in the program. Further, some projects showed evidence of sequence and selection only, but no loops for the concept of structured programming, while a few samples did not make use of the key data structures (struct, files and arrays).

Candidates are reminded to print their source code directly from the compiler as transferring to a word processor changes the spacing of the code. This may lead to candidates losing marks for appropriate programming style and documentation.

Generally, most candidates did not include adequate comments at key areas of their code and a few were not properly indented. However, this section was done well by most candidates.

Evidence that Code Matches Algorithm

This section for this was well done as most candidates were able to obtain full marks.

Evidence of FILE Manipulation

Candidates were required to provide a code that included FILE and showed evidence of FILE manipulation (open file, write to file, read from file, append file and close file).

Most candidates demonstrated evidence of this. However, some candidates were awarded marks by the teacher even though there was no evidence of FILE manipulation, and in some cases the use of FILE was non-existent yet marks were awarded.

Testing and Presentation

This year the majority of samples provided a suitable range of test data, but a few candidates

- did not have a test plan, but had screen shots only
- did not include all four testing criteria (normal, extreme, erroneous and incomplete) in their test plan
- had test results but did not have a clear test plan
- were awarded marks but had no test results
- did not include actual screen shots of the working program and testing was not done using the test data outlined in the test plan.

In one example a candidate was awarded marks for test results, when the candidate had not produced a test plan or any results. Therefore, the teacher might not have understood the marking criteria or did not understand what a test plan should be like.

Generally, the SBA projects this year were well presented. Most candidates gained marks between 41 and 50, followed by marks between 51 and 60, then marks between 31 and 40 marks and few candidates gained under 20 marks .This demonstrated that most SBAs were satisfactorily done.

Recommendation

Teachers need to ensure that the candidates use the headings outlined in the criteria for marking the School-Based Assessment Project in the syllabus, and that they also follow the order in which these headings occur. They should also ensure that candidates check that the numbering in their Table of Contents corresponds to the numbering in the body of the document and that candidates provide information in a logical way, using correct grammar and appropriate jargon at all times in the presentation of their projects. Teachers also need to use the correct forms recommended by CXC to mark and/or submit candidates' marks.

UNIT 2 – FURTHER TOPICS IN COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of 64.26 out of 90, a standard deviation of 13.57 and scores ranging from 22 to 90.

Paper 02 – Essay Questions

Section A – Data Structures

Question 1

Part (a) required candidates to suggest possible data structures that would fit applications that were described. Most candidates were able to score full marks on this part of the question.

Part (b) expected candidates to insert a string value into a linked list node and insert the node at the beginning of a linked list. Very few candidates scored full marks on this part, due to failure to place the string value in a linked list node, and failure to execute the pointer manipulations required to place the node at the beginning of the list.

Part (c) required candidates to diagrammatically represent a queue as data is inserted into it and deleted from it. Most candidates answered this part properly. Some candidates, however, confused the operation of a stack and a queue.

Part (d) expected candidates to describe three operations of a stack. Many candidates scored full marks on this part of the question; however, an inordinately high number described an operation not in the set of operations for a stack, such as a display operation that shows all the contents of the stack.

Part (e) required candidates to reverse a queue. Few candidates scored high marks on this part of the question, and those with better scores used an auxiliary structure or a swap to achieve this objective.

<u>Question 2</u>

Part (a) tested candidates' ability to perform linked list manipulations that delete data. This question was generally poorly answered. A large number of candidates did not appear to comprehend the importance of preserving a pointer location, and then executing pointer manipulations to achieve an objective.

Part (b) (i) tested the ability to declare an array of a specified size, and was fairly well answered. Most candidates scored full marks on this question, but there was a minority who specified an incorrect array size or had syntax errors.

Part (b) (ii) required that candidates populate the array with user-entered data. A high number of candidates scored full marks on this section. Some candidates, however, presented solutions with errors in the loop condition, or syntax errors while reading data.

Part (b) (iii) required candidates to write a program that uses a sequential search to look within the array for a user-entered value. Most candidates provided a fairly good answer to this question; however, common problems included errors in the loop condition, errors in reading the user value, or errors in output. A small number of candidates provided a solution that was not related to the sequential search.

Section B – Software Engineering

<u>Question 3</u>

For Part (a) candidates were expected to state two factors for risk assessment in software development, apart from time and costs. Most candidates answered this part of the question correctly, but a few provided only one appropriate answer.

Part (b) required candidates to describe CASE tools. Few candidates scored full marks on this part. Some did not mention that the tools were computer based, or that the tools were designed to aid the software development process, or explain the acronym.

Part (c) asked candidates to describe four weaknesses of the waterfall model. Few candidates scored full marks on this part. Common mistakes included repeating points in the response, and simply stating a property of the waterfall model without highlighting why that point could be a weakness.

Part (d) asked candidates to explain two weaknesses of rapid prototyping. Candidates who scored full marks stated two valid weaknesses with supporting information. Other candidates omitted supporting information, stated a property of rapid prototyping without stating why it was a weakness, or repeated a previously stated point.

Part (e) assessed candidates on the ability to draw a Level-0 DFD to describe information flow in a school registration system. A large number of candidates provided satisfactory responses; however, common errors included incorrect symbols to represent sources, data stores and processes, use of verbs for data flows, and nouns for processes, and neglecting the flows between processes as described in the registration system.

<u>Question 4</u>

This question tested candidates' knowledge of software development and database design. Part (a) required that candidates define the term 'cardinality' and provide one example. Whereas many candidates were able to give an appropriate example of cardinalities, that is, one to one, one to many, and many to many, the large majority of candidates failed to give an appropriate definition. Cardinality is the number of instances of an entity that may exist between two related entities.

Part (b) required that candidates list three characteristics of a well-engineered software product. This question was generally well answered with candidates noting that a well-engineered software product is maintainable, dependable, efficient, usable, portable, scalable and appropriately documented, among other qualities.

In Part (c), candidates were asked to differentiate between functional and non-functional requirements and to give one example of each term. The responses to this question were fair; many candidates were correct in distinguishing that functional requirements describe what the system should do but many candidates were not clear in stating that non-functional requirements are requirements that judge the operations of a system or constraints on how the system should work. In many cases, correct examples were not given to support the given definition. A good example of a functional requirement is that the system should reject character entries in the data field. A good example of a non-functional requirement is the response times the system must guarantee.

Part (d) asked that candidates explain two types of tests which could be completed during the testing phase, other than white box and black box testing. Most candidates attempted this question and correctly noted various types of tests, namely unit testing, which is used to validate that individual units of code or modules operate as intended, and integration testing, which combines software modules as a group to ensure that the program operates as intended. Other types of tests include stress/performance testing, recovery, usability, alpha testing and beta testing.

In Part (e), candidates were required to draw an entity relationship diagram for a given scenario related to a university called the Science College. Whereas weaker candidates did not score high marks in Parts (a) through (d), many were able to gain reasonable marks in this part of the question. In general, it appears that candidate performance on this topic is showing improvement based on the comments in the report for 2015. Once again common errors included the following:

- Pluralisation of individual entity names; **LECTURER** not LECTURERS, **COURSE** not COURSES
- Inconsistent ERD notation denoting cardinalities; Crow’s Foot or UML accepted.

Section C – Operating Systems and Computer Networks

Question 5

This question tested the candidates’ knowledge of types of networks, network architectures, the OSI model, routers and GPRS. Part (a) required candidates to describe the characteristics of coaxial cable and fibre-optic cable transmission media. Physical characteristics were the main focus here. For coaxial cable, there is a single centre copper wire symmetrically surrounded by a braided or foil conductor while fibre-optic cables consist of a bundle of extremely thin tubes of glass transmitting optical signals in the form of light. Though several candidates attempted this question, many opted to compare the given terms as opposed to what was required.

In Part (b), candidates were asked to explain what an IEEE 802.16 network is. In general, this part of the question was poorly answered. Many candidates appeared not to understand that this type of network caters for long-range wireless, broadband networks. Though several candidates noted that this type of network allowed for wireless communication, this was not sufficient as the notion of long-range wireless, WAN or MAN was important to differentiate the 802.16 standard from other wireless network standards.

Part (c) required candidates to explain the role of a switch in a local area network and accompany this explanation with a diagram. Many candidates attempted this question and were able to draw the supporting diagram correctly. Candidates lost marks for failing to note that the switch acts as a connectivity device on the network and sends signals from node to node efficiently without broadcasting messages to every node on the network.

Part (d) required candidates to describe how data is communicated in an IEEE 802.11b network, using a diagram to support the response. Candidates were required to note that the medium for transmission is wireless. Nodes must first associate with the access point, then for a node to transmit to another node the communication must pass through the access point. Though in many cases wireless transmission was noted, the details regarding the access point were omitted. As far as the diagram was concerned, candidates were expected to clearly depict the access point along with the wireless node while using the appropriate symbol to represent wireless communication.

In Part (e), candidates were asked to briefly explain specific factors that should be considered when building a network. These factors included cost, security, management, expandability and interconnectivity. Though many candidates attempted this question, several candidates did not clearly expand on the given factors. For example, where security is concerned, issues of encryption, protection of data and resources, firewalls, and virus protection are some of the detailed considerations, while in terms of management, it is important to oversee the network and fix the problem while routinely maintaining the network. Many candidates did not provide correct responses where expandability and interconnectivity were concerned. Expandability is concerned with ensuring that the addition of new nodes is easy while interconnectivity ensures that connecting to other networks and the Internet is seamless.

Finally, Part (f) dealt with the OSI model for network communication and asked that candidates explain the role of any two of the first five layers of this model. Correct responses would include the physical, data link, network, transport and session layer. Though this question was generally well answered, some candidates only noted the actual layer without a supporting explanation. This indicated that the purpose of the layer was not clearly understood.

Question 6

This question tested the candidates’ knowledge of operating systems, namely, memory management, device management, interrupts, types of operating systems, file security and the process control block. Part (a) required candidates to explain the terms ‘virtual memory’, ‘paging’ and ‘thrashing’. This question was poorly answered. Candidates were expected to note that virtual memory is disk space used to maximize the amount of RAM available to programs, while paging is a memory management scheme used by operating systems where data is copied from secondary storage in blocks for use in main memory. On the other hand, thrashing occurs when a computer’s virtual memory subsystem is in a constant state of paging, rapidly exchanging data in memory for data on disk.

Part (b) required candidates to explain how excessive paging affects a computer. Many candidates attempted this question and were able to state that excessive paging causes degradation in the computer’s performance. However, many candidates did not note that this degradation could lead to the collapse of the system.

Part (c) tested candidates’ knowledge of device drivers in an operating system. Though this question was widely attempted, it was not always clear that a device driver contains a special code which the operating system uses to connect to the device.

Candidates were provided with a short scenario in Part (d) and asked to explain how the operating system uses interrupts in the given scenario. Few candidates identified the type of interrupt that occurred as an i/o interrupt though many were able to describe the subsequent context switch, reading from the file followed by another interrupt in order to continue the previous process.

In Part (e) candidates were asked to distinguish between a multi-user system and a batch processing system. This question was generally well answered, with candidates stating that a multi-user system is a computer with an operating system that supports multiple users at once or at different times. In contrast, jobs can be stored/accumulated and later executed in batches in a batch processing system.

In Part (f), candidates were asked to outline one way, besides passwords, in which files can be protected. Several candidates correctly noted encryption or the encoding of data in their solution while other candidates noted access control lists which specify permissions for users. Though lock words were also noted, these are recognized as a type of password.

Finally, in Part (g), candidates were asked to identify three components of a process control block. Many candidates answered this question successfully by noting the process state, process ID, program counter, CPU registers, memory management information and accounting information as components of a process control block.

Recommendations

- This subject is now marked using the on-screen marking system and as such when scripts are scanned, care is taken to scan **only** those areas provided for writing. Teachers should emphasize that candidates should write their solutions **only** in the boxes/lines provided. Where this space appears insufficient, the extra pages provided should be used. Candidates should avoid writing in the white spaces surrounding the boxes/lines provided, as there is a risk of truncation during the scanning exercise.

- Candidates’ ability to express themselves clearly in writing is at times weak. This makes the process of assessment difficult. Candidates should be mindful that clear and succinct answers are best, ensuring that the solution matches the question posed.

- Teachers should ensure that candidates are fully prepared for the examinations in both units. The poor performance in some modules of the syllabus indicates that more time is needed or that different pedagogical strategies should be utilised when teaching these subject areas.

Paper 03 – School-Based Assessment (SBA)

Each candidate was expected to choose a problem for which a software solution exists and then use software engineering techniques to develop the software. In particular, the candidate was expected to demonstrate appropriate choice of the tools and techniques used in the analysis of the software to be developed. They were then expected to design, code and test their software, using appropriate techniques.

General Comments

- Candidates should follow the order and the headings laid out in the criteria for marking when compiling the SBA report.
- Candidates should avoid using water marks when printing the SBA as it makes it difficult for the examiner to read and assess their work.
- Teachers should avoid using red ink pens to correct SBA.
- Teachers should ensure that each SBA is clearly labelled with the candidate's name and centre number.

Marking Criteria

1. Specification of Requirements

- *Definition of Problem:*

Candidates were required to give a complete and accurate description of the problem. Most candidates handled this section fairly well but there were a few candidates who were still unclear of what to include in their definition of the problem. A brief description of the context in which the problem occurred is required but details about actual problems staff and/or clients face and proposed steps to correct such problems should be emphasised. Most candidates focused on the proposed system instead of adequately describing the problem to be solved, while some omitted a proposal altogether.

- *Techniques of Analysis:*

Candidates were required to identify techniques of data collection and to analyse the data collected.

i. Some candidates were able to state the various techniques of data collection; however, most candidates did not provide a justification for their selection. Those candidates misinterpreted justification to mean definition. Marks are not awarded for definitions. Candidates should clearly explain why the chosen technique was used as it relates to the organisation and should not regurgitate advantages of the technique in order to gain marks.

ii. Proof of analysis should also be given. All proof must be included **directly** after the analysis and **not** in the appendix. For example, sample questions from the questionnaires and/or interviews should be included. It is understood that it may be difficult to include proof of observation; hence, if this technique is used, candidates should clearly describe what was observed.

- *Use of Data Flow Diagrams and E-R Diagrams*

This section of the School-Based Assessment has improved significantly. Candidates were able to properly draw relevant diagrams that correctly mapped to their problem statement.

Context Level Diagram

Candidates were required to give a complete and accurate diagram of all relevant entities and data flows. The weaker candidates generally failed to label their data flows.

Level 1 Diagram

Candidates were required to give a complete and accurate diagram with all relevant processes, data flows and major data stores.

In this regard, the following points are to be noted:

i. Only one level 1 diagram is required.
ii. Most candidates are unaware that the level 1 diagram is an expansion of the context level diagram. Hence, new external entities were created for the level 1 diagram, and new data flows were created for previously used entities.
iii. Candidates had links between data stores and entities; data stores and data stores; entities and entities, which are **all** incorrect.
iv. Some candidates' diagrams were too small, which made reading and marking difficult.

<u>ER Diagram</u>

Candidates were required to give a complete and accurate diagram of all relevant entities and relationships.

i. Most candidates did use the correct symbol for a relationship (a diamond). Nevertheless, there were still candidates who did not use a symbol but wrote on the line instead. This is not acceptable.
ii. Some candidates did not include attributes for the entities.

Functional and Non-Functional Requirements:

Candidates were required to give a complete and accurate description of all requirements.

i. For the functional requirements, candidates did not clearly state what the system is supposed to do. Instead they stated what the user would be doing. An example of a good functional requirement is *the system will be able to delete a patient's record.*
ii. For the non-functional requirements candidates did not state the limitations of the system. An example of a good non-functional requirement is *the system can only store 1000 patient records*.
iii. Candidates are encouraged to use bullets or number their functional and non-functional requirements.

2. <u>Design Specification</u>

Candidates were required to give a complete and accurate system structuring diagram containing all processes and a description of the user interface, report design, algorithm design and appropriate data structures. Most candidates handled this section fairly well.

i. Most candidates were unaware of how to correctly draw a system structuring diagram. Candidates should note that this diagram is similar to the HIPO chart as it shows the breakdown of the system into its submodules and the submodules into its further components.
ii. Candidates should not just include screens shots of interface and report design but also a justification in order to gain maximum marks.
iii. Narratives will **not** be accepted as an algorithm. Candidates are reminded to submit either pseudocodes or flow charts. They must **not** submit both.

3. <u>Coding and Testing</u>

Candidates were required to produce a complete and accurate C program solution for the problem stated in the Definition of Problem stage. Most candidates produced programs that achieved good functionality.

i. Only procedural C is accepted as the programming code. No other programming language will be accepted. Code must also be printed from the compiler, **not** transferred to a word processor before printing.
ii. Candidates should **not** include soft copies of their School-Based Assessment report. Soft copies will NOT be marked.
iii. Most candidates did not include enough screen shots and in some cases no screen shots were included. Since a soft copy is not required, the screen shots are very important in allocating marks for the functionality of the code.
iv. Some codes presented did not match the screen shots given. In such cases candidates were not awarded any marks.
v. Test plans should be written in a tabular format. They should include normal, extreme and erroneous.
vi. **All** input data must be tested using the three testing criteria mentioned above.
vii. Test results submitted without related test plans will **not** be awarded any marks.

4. <u>Communication and Presentation</u>

This section of the School-Based Assessment is often overlooked by candidates. Teachers and candidates are urged to pay close attention to their use of grammar and the overall presentation of the SBA report. Candidates often lose marks due to poor sentence structure and failure to use appropriate technical jargon in their reports

Recommendations

Teachers should ensure that candidates are fully prepared for the examinations in both units. The poor performance in some modules of the syllabus may indicate that more time needs to be allocated to these areas. Encouraging students to review this subject report can be very helpful in improving performance.

C A R I B B E A N   E X A M I N A T I O N S   C O U N C I L

REPORT ON CANDIDATES' WORK IN THE
CARIBBEAN ADVANCED PROFICIENCY EXAMINATION®
MAY/JUNE 2017

COMPUTER SCIENCE

GENERAL COMMENTS

This was the ninth year in which the revised syllabus was examined. There were three examination papers in each of Units 1 and 2, namely, Paper 01, Paper 02 and Paper 03. In each unit, Paper 01 and Paper 02 were examined externally by CXC while Paper 03, the School-Based Assessment, was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of 45 multiple-choice items that were designed to test candidates' breadth of coverage of the syllabus. Paper 02 consisted of six essay-type questions that were designed to test candidates' depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

The individual contributions of Paper 01, Paper 02, and Paper 03 to the final grade are 30 per cent, 50 per cent, and 20 per cent, respectively.

Approximately 90 per cent of the candidates obtained Grades I–V in Unit 1 and approximately 93 per cent of the candidates obtained Grades I–V in Unit2. Overall, there is still need for improvement in the quality of responses for the programming questions in both units.

DETAILED COMMENTS

UNIT 1 – FUNDAMENTALS OF COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of approximately 60 out of 90, standard deviation of 15.39 per cent and scores ranging from 14 to 90.

Candidates performed well across Module 1 (Computer Architecture and Organisation) and Module 2 (Problem Solving with Computers) but poorly across Module 3 (Programming).

Paper 02 – Essay Questions

Performance on the essay response questions was reflected in a mean of approximately 54 out of 150, standard deviation of 26.87 and scores ranging from 2 to 132.

Section A – Computer Architecture and Organization

Question 1

This question focused candidates' attention on data representation in computer systems and low-level computer organization. The maximum score on this question was 22/25, and the mean score was 11.4. Almost 15 per cent of candidates scored below 5 marks and less than 25 per cent scored above 15 marks. These statistics signify basic knowledge of the concepts under investigation.

Part (a) (i) required candidates to write down the range of decimal values that can be represented. A large number of candidates obtained full marks for this part. Most candidates were able to provide the largest positive integer, namely +127. However, some candidates lost the mark for stating –127 as the largest negative integer. Very few candidates accurately stated –128 as the largest negative integer. A few candidates converted the bit string 01111111 and stated +255.

Part (b) required candidates to convert a number stored in a floating point format to a decimal representation. The majority of candidates was able to correctly decompose the binary number into sign, exponent and mantissa. Approximately 25 per cent of the candidates gave accurate responses, and showed their working. About ten per cent of candidates showed evidence of understanding the decomposition process, with errors in the conversion process. Some candidates were unable to convert the fractional part of the binary number to decimal to obtain the correct answer. A large number of candidates did not show evidence of the decomposition.

Part (c) required candidates to draw the truth table for the logic expression x.y' that is, x AND NOT y). This part was well done by most candidates, with many gaining full marks. A few candidates lost marks for incorrect combinations and failure to include column headings in the table. Candidates are encouraged to be systematic as they provide the various combinations for the inputs, that is, maintain order from binary 0 to 3.

Part (d) required candidates to draw a circuit that can accept two binary inputs, referred to as x and y respectively. The circuit should represent the propositional logic for x AND NOT y.

Most candidates were able to answer this question correctly identifying the logic gates to be used. The main weak area in response to this part was the wiring of the circuit.

Part (e) required candidates to describe the function of the following circuits in a computer:

- (i) Adder
- (ii) Shift register
- (iii) Counter

This part was poorly done by approximately 95 per cent of the candidates. Very few candidates demonstrated understanding of the function of the circuits. Most candidates were unable to describe the function of any of the three circuits identified and simply indicated that the circuits added, shifted and counted respectively. A few candidates stated that the counter was used to store the address of the next instruction to be executed.

Part (f) required candidates to draw a clearly labelled diagram of a multiplexor that selects one output from four possible data inputs.

This part was generally well done by most candidates. The labelling of the input lines and the selection lines was the main area of concern. This resulted in some candidates losing one or two marks in some instances. In some cases, candidates drew circuit diagrams inclusive of various logic gates, when in fact the question required only a block diagram.

Part (g) required candidates to describe how a multiplexor can be used to select the correct output in an ALU that contains circuitry to execute the functions AND, OR, addition and shifting.

This part was not well done by the majority of candidates, with approximately 25 per cent providing responses that gained the two marks allocated. The majority provided responses that did not demonstrate understanding of the question. Candidates were penalized for failure to apply use of the multiplexor to the given scenario.

Part (h) required candidates to state the number of selection control lines that are needed in a multiplexor that has ten inputs. This part of the question posed a challenge to candidates since most were unable to provide the correct answer. Most candidates appeared unable to calculate the number of selection control lines for a given number of inputs.

Question 2

This question assessed candidates' comprehension of high level computer organization. The highest score on this question was 18/23. Approximately 30 per cent of candidates scored below 5 marks and 31 per cent scored over 15 marks. These statistics signify that many candidates lack basic understanding of the topics. While some candidates had an understanding of the topics, there was no demonstration of mastery.

Part (a) tested candidates' ability to use a diagram to show how a set of flip-flops can be used to create a 4-bit register. Approximately 25 per cent of candidates presented properly labelled diagrams with parallel input and output lines. Many candidates presented diagrams that did not illustrate that the set of flip flops are loaded and accessed in parallel. A large number of candidates did not seem to appreciate that four flip-flops were required to construct the register. Some candidates did not seem to interpret the question properly and instead presented diagrams that attempted to represent the internal components of a flip-flop.

Part (b) required candidates to describe the purpose of a decoder while accessing RAM. Only a few candidates properly described the function of a decoder in selecting an appropriate memory location as an output when an address is applied to its input. A few candidates properly described the function of a decoder, while ignoring its application with memory. Common responses included the use of a decoder to decode the contents of memory or as a part of the instruction cycle.

Part (c) required candidates to derive the maximum number of RAM locations that can be accessed in a computer that uses an 8-bit memory address. Most candidates properly showed that the number of accessible locations was 28, giving 256 locations. Other candidates presented 255 as an answer. A few candidates presented other responses.

Part (d) required candidates to explain why data stored in memory registers is accessed faster than data stored in RAM. Only a few candidates presented the required points, namely that registers are included in the processor, and can therefore be accessed directly, and that additional steps are required to copy data from main memory to registers. Most candidate responses contained a part of the answer, and a few contained answers that were not relevant.

Part (e) required candidates to name three phases of the instruction cycle and outline how each phase operates. A large number of candidates stated the expected three phases; however, only approximately 75 per cent of those who selected the phases provided outlines for all three expected phases. A small number of candidates presented responses that did not correspond with any phase in the instruction cycle.

Part (f) required candidates to discuss why a cache is expected to speed up a computer system. Approximately 40 per cent of candidates scored three out of the four allotted marks by pointing out that cache is a high speed memory buffer that stores frequently executed instructions. Most candidates gave a part of the answer. A few candidates gave responses that described the function of an application cache. A few candidates gave responses that were irrelevant.

Part (h) required candidates to state two examples of information that is stored in ROM. Approximately 30 per cent of candidates provided two valid responses, and approximately 70 per cent provided at least one valid response. A large number of candidates included responses such as operating system and application data.

Section B – Problem Solving with Computers

Question 3

This question tested candidate's ability to interpret and create algorithms. Two candidates achieved full marks on this question, and the mean score was 13.59. Forty-four per cent of the candidates scored above 15 marks and approximately ten per cent of candidates scored less than five marks.

Performance on this question indicated that many candidates were comfortable with algorithmic concepts and that there was some mastery of these concepts.

Part (a) required candidates to identify three properties of well-defined algorithms. A large number of candidates gave appropriate responses. Candidates also demonstrated the ability to terminate and possess a logical flow of control. Approximately 50 per cent of candidates gave two out of three appropriate responses, and most provided at least one out of three responses. Other candidates provided responses such as 'Input, processing, output' or listed a set of programming control structures.

In Part (b), candidates were expected to describe the problem analysis stage of the problem-solving process. Very few candidates presented the expected response that additional information gathering strategies, or decomposition, is used to determine the root causes of the problem. Some candidates provided a part of the expected response, but many of them associated problem analysis with a solution generation strategy.

Part (c) presented candidates with a scenario, and expected them to prompt for a temperature and read it, then perform a calculation and conditionally provide output. Most (approximately 75 per cent) candidates provided a valid response worthy of full marks. Some candidates made minor errors such as omitting the read instruction, calculate incorrectly, or writing the conditional output. There were a few candidates who provided inappropriate responses such as a series of print statements.

Part (d) presented and algorithm, for which candidates were expected to write the output. Approximately 40 per cent of candidates wrote the expected output. One common mistake was the presentation of only one output, when two were expected for a given value of num. Another common mistake was the inclusion of quotes, or additional output such as 'The output is ....'.

Part (e) required candidates to write an algorithm to handle a case that involved a specified number of repeated inputs that were to be categorized and tallied, and a decision made based on the result of the tally. Approximately ten per cent of candidates scored more than 19 per cent of the allotted marks. Common mistakes included omitting initializations, omitting the loop, neglecting to terminate the loop, illogical conditions and outputs that did not meet the specification.

Question 4

This question assessed candidates' command of basic control structures. Ten candidates achieved full marks on this question, and the mean score was 10.84. Approximately 30 per cent of the candidates scored 15 marks or higher, and approximately 26 per cent of the candidates scored 5 marks or less.

Performance on this question indicates that while some candidates have mastered the topics, there is a significant number of weak candidates.

Part (a) required candidates to list three types of control constructs. This part was generally well done. In a few cases, examples were provided by candidates, for example, *if statement* instead of *selection control construct*.

Part (b) required candidates to write an algorithm to determine the number of 12-slice pizzas a teacher would have to purchase in order for each candidate in his class to receive one slice. The algorithm should also calculate and print the number of slices remaining. This part was fairly well done by most candidates. Most of the responses attracted a score of three out of five. Candidates lost marks as a result of assuming a number of candidates, for example, 30 and inserting that value into the algorithm, rather than developing a solution to work for all cases.

Part (c) required candidates to update a given algorithm to allow it to use bounded iteration to find and print the sum of the squares of all numbers between 0 and 1000 inclusive.

For this part, approximately 50 per cent of candidates gave correct answers and 50 per cent incorrect answers. Many candidates failed to initialize the sum variable. There was evidence of the use of the looping construct but some candidates failed to score marks for limits and the sum of squares.

Part (d) required candidates to write an algorithm to accept marks for a written and oral exam for 30 candidates, and to calculate and print the number of candidates who passed each exam as well as the average for all candidates in each exam.

This part was fairly well done. Most candidates were able to gain the marks for initializing variables and reading the marks for the written and oral exams. Candidates generally demonstrated understanding of the logic of the solution. However, in some cases, candidates' algorithms added the written mark to the oral mark and then checked whether this total was greater than 50 and incremented counters based on combined marks. Candidates then printed from two variables instead of four.

Section C – Programming

Question 5

This question tested candidates' capability to interpret and use function calls in C, and to manipulate complex data structures. One candidate achieved full marks on this question, and the mean score was 3.79. Approximately five per cent of the candidates scored 15 marks or higher and approximately 78 per cent of the candidates scored five marks or less. It should be noted that of the 246 candidates who scored zero on this question, a large number did not attempt a response. Also, a large number of candidates did not attempt the subparts within the question.

Performance on this question indicates that a small number of candidates mastered the topics, while a large number has a poor grasp of the content.

Part (a) (i) presented candidates with an algorithm, and asked them to state the results of tracing the algorithm and reporting the results at the beginning of each iteration. Approximately 25 per cent of candidates gave an appropriate response, while approximately 50 per cent of candidates wrote a response that stated 4 was the first value of the variable result. Most of the other responses were incorrect.

Part (a) (ii) presented candidates with a function that executed a mathematical operation, and asked for an expression that applied the operation in terms of the function. Approximately 15 per cent of candidates responded with appropriate answers. About ten per cent had appropriate calls to the *fact* function, but with logical errors in the calculation. The other candidates did not present solutions that called to the *fact* function. A mark was allocated to an appropriate call and specification of a return value, and most candidates earned that mark.

Part (b) tested candidates' capability to explain how the assembly and linking process allows code written in high level programming languages to run on a wide range of machines. The expected response that pre-compiled code, some of which is machine-specific, is combined with other modules to create executables was received from only a few candidates. Most candidates scored two out of three marks for describing the combining of pre-compiled code, but seemed to fail to appreciate that machine language was specific to architectures. There were a few candidates with responses that did not answer the question.

Part (c) tested the candidates' capability to manipulate a complex data structure that consisted of an array of records.

Part (c) (i) expected candidates' to initialize a counter variable loop once per month through a section of the data structure referenced by *islandIndex*, and increment the counter variable for any entry that meets the validity condition. Approximately 20 per cent of candidates scored full marks on this part of the question, with most of the others suffering from errors either in initialization, looping to the incorrect limit, or in referencing the data structure. A large number of candidates did not attempt the question.

Part (c) (ii) asked candidates to write a C module *avgTemp (intislandIndex)* that evaluates the average of all valid readings. Approximately 10 per cent of candidates presented code that could accurately record the number of valid readings as well as the sum of the valid readings, allowing the average to be calculated. Most candidates ignored the fact that some readings were valid, and provided a result that was equal the sum of the readings divided by the number of months. A large number of candidates did not attempt the question.

Part (c) (iii) requested that candidates write a main function that iterates over all records in the structure caribHistory and prints a report with the location. Approximately five per cent of candidates returned good responses that iterated through locations, checking validity of the location's data and printing location name and average. Common errors in other responses included an iteration through months instead of locations, an invalid validity condition (the number of valid readings must be greater than 6), referencing errors with the data structure, and invalid format strings ("per cents") for strings and "per centf" for floats. A large number of candidates did not attempt a response.

<u>Question 6</u>

This question assessed candidates' understanding of programming practices and their ability to interpret and write simple programs.

Three candidates achieved full marks on this question, and the mean score was 6.84. Approximately 14 per cent of the candidates scored 15 or higher and slightly over 51 per cent of the candidates scored 5 marks or less. While these statistics are better than those gleaned from the previous question, they underline the point that programming knowledge is generally well below par.

Part (a) required candidates to list two ways in which good programming style can be maintained and explain why each way is important. This part was well done by most candidates. Most candidates were able to list two ways and give correct explanations. The other candidates for the most part, were able to state two ways but explanations were vague.

Part (b) required candidates to list three stages in the translation process. This part was well done. Very few candidates were unable to correctly list at least two of the stages.

Part (c) required candidates to write a C function which accepts an integer array and an integer variable indicating the size of the array. The function should return the location of the maximum value in the array.

This part was generally well done. Most candidates were able to correctly declare the function header. Even though many candidates failed to initialize the location variable, they traversed the loop with correct condition. Some candidates tracked and returned the maximum value rather than the location of the maximum value.

Part (d) (i) required candidates to write a C function *getZoneFee* that accepts a customer's ID and returns the flat fee to be charged, based on the scenario in the question. Most candidates were able to declare the function header correctly. In some cases, Customer ID was not declared as a parameter and candidates' functions then read the Customer ID from the keyboard. Most candidates used a series of if statements rather than a switch construct, and of the few who used the switch, most referenced the second location in the array (*customer_id[1]*) rather than the first (*customer_id [0]*).

Part (d) (ii) required candidates to write a C function that accepts a customer's ID and the weight of a package and returns the delivery fee, using the function from Part(d) (i). This part was not well done by the majority of candidates. Few declared the function header with appropriate parameters. Many candidates were also unable to call the *getZoneFee* function. However, most candidates showed evidence of multiplying the weight by 5 in their functions.

Part (e) required candidates to show the output generated by a given segment of code. This part was poorly done by the majority of candidates. Most candidates were able to get the first line of the output correct. Few, however, were able to print the other lines correctly as some candidates did not understand the code and/or printed on the same line or did something unrelated to the code.

In all items requiring C code, candidates, for the most part, had the logic in place but were challenged by the syntax of the language. Teachers are encouraged to ensure that students become more familiar with the programming language.

Paper 03 – School-Based Assessment (SBA)

General performance was reflected in a mean score of 47 out of 60, with a standard deviation of 9.55 per cent and scores ranging from 2 to 60.

UNIT 2 – FURTHER TOPICS IN COMPUTER SCIENCE

Paper 01 – Multiple Choice

Performance on the 45 multiple-choice items on this paper produced a mean of approximately 63 out of 90, standard deviation of 13.76 per cent and scores ranging from 20 to 88.

Candidates performed well across Module 1 (Data Structures) and Module 3 (Operating Systems and Computer Networks). Candidates' performance was moderate across Module 2 (Software Engineering).

Paper 02 – Essay Questions

Performance on the essay response questions was reflected in a mean of 53 out of 150, standard deviation of 24.82 per cent and scores ranging from 1 to 126.

Section A – Data Structures

<u>Question 1</u>

This question tested candidates' capability to interpret and write algorithms that perform searching and sorting operations over specified data structures. The mean score was 7.84, and the highest score of any candidate was 23. Ten per cent of the candidates scored 15 marks or higher and approximately 40 per cent of the candidates scored 5 marks or less.

Part (a) (i) required candidates to write a series of steps to describe how the binary search algorithm will search an unsorted array for a given key.

This part was fairly well done. Most candidates demonstrated knowledge of how the binary search algorithm works. However, many did not relate their steps to the array as given. Too many candidates indicated that the search algorithm would first sort the items in the array. Most candidates scored three marks for this part.

Part (a) (ii) required candidates to state, with supporting explanation, whether the search algorithm returns the correct location. Few candidates gave the correct response. Since many candidates indicated that the array would first be sorted, they stated that the search algorithm would return the correct location. In many instances, candidates stated that the algorithm would not return the correct location, but their explanations were not correct.

Part (b) required candidates to use the bubble sort to sort a given array in ascending order, drawing the array after each complete pass of the bubble sort. This part was generally well done. In a few cases, candidates drew the array after each sub-pass and were not sure where the complete passes ended. Many candidates simply used any 'method' to sort the array ending with a sorted array with no relation to the bubble sort algorithm.

Part (c) (i) required candidates to draw a singly linked list with a single node containing the value 8. This part was well done. However, some candidates did not show the null pointer (pointer to null). Part (c) (ii) required candidates to use a diagram in explaining how the number 3 could be added to the beginning of the linked list. For this part, most candidates were able to draw the diagram correctly. In their explanations, however, candidates' manipulation of the pointers led to the loss of the list, that is, candidates did not ensure that there was always a pointer to the head of the list.

Part (d) required candidates to write C code to sort an array using simple selection sort. This part was well done. Candidates demonstrated a good understanding of how this algorithm works, showing the need for the use of a temporary variable in making the exchange at the end of a pass.

<u>Question 2</u>

This question validated candidates' capability to manipulate stack and queue abstract data types. The mean score was 8.75, and one candidate scored full marks. Approximately ten per cent of the candidates scored 15 marks or higher and approximately 38 per cent of the candidates scored 5 marks or less.

Part (a) tested candidates' ability to apply stack and queue ADT operations to achieve the objective of reversing the contents of a stack. The expected solution that iteratively pops from the stack and enqueues the result until the stack is empty, then iteratively dequeues and pushes the result to the stack until the queue is empty. Pops from the queue was supplied by a relatively small number of candidates. Common errors included logical errors in the process, or the use of unsanctioned data structures. Logical attempts that did not refer to the sanctioned ADTs earned partial credit. Many responses made reference to array operations that did not lead to a solution.

Part (b) presented candidates with a stack and some operations on the stack, and required candidates to draw the stack after each of the operations were executed. Most candidates scored full marks on this question. Some candidates did not show full working, and responses from a small number of candidates showed one operation affecting more than two elements, demonstrating a lack of understanding of stack operations.

In Part (c) (i), candidates were expected to declare variables required for the implementation of a queue in C and describe the purpose of each variable.

While more than 50 per cent of candidates scored full marks on this question, common mistakes included neglecting to declare the memory structure that can hold the data, and variables designated to reference the front and rear of the queue. A large number of candidates failed to earn allotted marks due to unclear descriptions of the purpose of variables.

Part (c) (ii) required candidates to write C code to implement enqueue and dequeue functions. Approximately ten per cent of candidates earned full marks for this section. Common errors included failure to test overflow and underflow conditions appropriately, failure to add to the queue at the rear and remove from the front, and failure to update front and rear reference variables. It was observed that two main removal strategies were employed across the workable solutions, namely, removal from the front and incrementing the front pointer, and removing the first element and shifting all elements forward.

Many responses (approximately 40 per cent) exhibited multiple instances of the errors indicated above.

Section B – Software Engineering

Question 3

This question tested candidates' ability to interpret and use software engineering tools. The mean score was 10.96, and one candidate scored full marks. Approximately 21 per cent of the candidates scored 15 marks or higher and approximately 17 per cent of the candidates scored 5 marks or less.

Part (a) required candidates to draw a Level-0 data flow diagram that depicts a given scenario.

This part was generally well done. Most candidates were able to score between 9 and 12 marks. The main areas of concern in candidates' responses were the convention used for representing flow chart symbols, and failure to recognize that process names and data flows are written in verb form and noun form respectively. Some candidates continue to:

- confuse Level-0 with context diagrams
- use inconsistent and incorrect symbols to represent sources, data stores and processes
- use verbs for data flows
- connect sources directly to each other
- draw processes with data in-flows only.

Part (b) required candidates to define the term *deliverable* in the context of the Software Development Life Cycle (SDLC). This part was generally well done. Candidates demonstrated that they knew what was meant by the term *deliverable*.

Part (c) required candidates to list two deliverables from the design phase of the SDLC. This part was generally well done. However, some candidates listed deliverables from other phases, for example, the analysis phase.

Part (d) required candidates to briefly describe how the fountain approach to software development is different from the waterfall approach. This part was generally well done. Most candidates were able to outline differences between the two approaches. A few candidates indicated that the only difference was that one was a top-down while the other was a bottom-up approach.

Part (e) required candidates to define prototyping and outline how it is useful in the software development process. This part was generally well done. Most candidates spoke to the idea of a model of the software developed early in the process and the main purposes being to have the client interact with the software and to provide feedback to the developers.

Part (f) required candidates to define a HIPO chart. This part was not well done. Most candidates were only able to score one mark for knowledge of what the acronym means.

Question 4

This question tested candidates' understanding of relationships between entities in systems and their ability to design tests. The mean score was 10.10 and the highest score of any candidate was 22. Approximately eight per cent of the candidates scored 15 marks or higher and approximately 16 per cent of the candidates scored 5 marks or less.

Part (a) required candidates to define *cardinality* and *modality* in the context of entity relationship diagrams. Approximately 25 per cent of responses satisfactorily defined cardinality as the number of times an instance in one entity can be associated with instances in the related entity. Some candidates scored a mark for stating an example of a cardinality without an appropriate definition. A large number of candidates simply defined cardinality as the *relationship between entities*, and that definition was not accepted.

The definition of modality as an indication of an optional or mandatory relationship was extremely rare across responses, and most candidates responded that the modality described the nature or purpose of the relationship between entities.

Part (b) required candidates to draw an ERD based on a given scenario. Most candidates used appropriate symbols; however, the following errors were common:

- Pluralisation of individual entity names: *Volunteers* instead of *Volunteer*
- Repeated relationships, for example, a relation from *Program* to *Volunteer* was presented along with a separate link from *Volunteer* to *Programme*

In Part (c), candidates were requested to describe three tests that could be used to validate the correct operation of a function. While candidates were expected to provide a description of the data used to test and the outcomes unit test strategies and descriptions were accepted. 'System testing' or 'integration testing' were not accepted as valid responses, as the question specified the unit to be tested. Approximately 50 per cent of candidates responded with test data and outcomes. Some of these, however, repeated the description of one test as a second, and some did not describe the expected outcome of the test. Other candidates specified a test in which one of the data types was incorrect, which was not accepted as the question specified the data types of the arguments.

Part (d) required candidates to explain the terms *usability, maintainability* and *portability*. The expected response for usability is easy, effective use of a system. A large number of candidates specified ease of use but few specified effective use. The expected response for maintainability was any two of the points of easy fault detection, system update, and ability to satisfy new requirements. Approximately 30 per cent of candidates scored full marks and another 50 per cent scored at least one mark. The expected response for portability was that the system should be able to run on a wide range of platforms. Responses that indicated that portability refers to the ability of a program to be copied were not considered a part of the answer. Responses that indicated the program operating on multiple machines without indicating multiple platforms or operating systems were not considered a part of the answer.

Section C – **Operating Systems and Computer Networks**

<u>Question 5</u>

This question tested candidates' knowledge of HTTP, processes and paging. The mean score was 8.23 and the highest score of any candidate was 22. Approximately 5 per cent of the candidates scored 15 marks or higher and approximately 30 per cent of the candidates scored 5 marks or less.

Part (a) requested that candidates state the name of the computing model in which HTTP is used. The expected response was the client-server model. However, the OSI networking model was accepted. More than 50 per cent of candidates provided valid responses; however, some of them supplied incorrect responses such as the waterfall model.

In Part (b), candidates were asked to describe the steps involved in HTTP when a user views a webpage using a web browser. The expected response that describes the resolution of the IP address, submission of the request to the server, return of the response and subsequent viewing in the client browser was described by only a few candidates. About ten per cent of candidates specified the steps that excluded resolution of the IP address, about 50 per cent of candidates supplied at least two of four of steps accurately, and approximately 80 per cent specified at least one step. Candidates who did not score marks supplied answers that did not include the role of HTTP, such as 'the link in blue is clicked so the user can view the page'.

Part (c) required candidates to demonstrate knowledge of whether http or https is preferred for online shopping. Approximately 90 per cent of candidates described the security features as an asset of the https protocol, but only approximately 15 per cent of those mentioned the encryption strategies that implemented security. Other candidates mentioned that https was more appropriate without supporting information, or did not indicate that https was more appropriate.

Part (d) asked candidates to state the relationship between a process and a process control block (PCB). Approximately 50 per cent of candidates provided the expected response that a PCB contains information about a process that is used by the OS to manage execution. A common response, however, was that a process control block 'controls a process'.

Part (e) asked candidates to list five possible states of a process. Most candidates scored full marks on this part of the question. A few candidates did not provide relevant states.

In Part (f) (i), candidates were required to define *paging*. Only a few candidates gave the expected response that paging is a virtual memory strategy that swaps equal sized blocks between the main memory and secondary storage. Approximately 70 per cent of candidates provided partial credit responses.

Part (f) (ii) required candidates to define *thrashing*. About 25 per cent of the candidates provided the expected response that thrashing is a state in which the effort of the processor in paging operations is so extreme that it severely hampers application performance. Approximately 30 per cent of candidates provided partial responses such as 'thrashing is excessive paging'.

In Part (g), candidates were expected to state two possible consequences of thrashing. Approximately 50 per cent of candidates gave the expected response that systems may slow down or crash. Eighty per cent of candidates gave at least one valid response. A frequently encountered response was 'The computer will overheat', for which no marks were awarded.

Part (h) asked candidates to give one reason why thrashing might occur. The full expected response that too many programs are being run on systems that do not have enough available RAM was provided by approximately 20 per cent of candidates. Approximately 50 per cent of candidates provided a part of the answer. Many of the other responses presented a consequence of thrashing as a cause.

<u>Question 6</u>

This question tested candidates' knowledge of network connectivity and interfacing between shared processes. The mean score was 7.18 and the highest score of any candidate was 21. Approximately 6 per cent of the candidates scored 15 marks or higher and approximately 40 per cent of the candidates scored 5 marks or less.

Part (a) (i) required candidates to demonstrate the ability to select an item of equipment that can extend internet capability to a home network, clearly explaining resulting interconnections with the aid of a diagram. In the expected response, a wireless router would be connected to the home devices and the modem, and the modem connected to the wireless router and the ISP. Approximately 40 per cent of candidates provided diagrams that described the scenario. However, the explanation of some candidates lacked details (such as a statement affirming that the router to be used is wireless), and clear statements of the connections to the router and to the modem. A common error was the placement of the desktop on a separate connection from the modem instead of from the router (effectively placing it on a separate network). Erroneous solutions that neglected to represent the ISP were also received.

In Part (a) (ii), candidates were required to identify the type of device needed to copy files between the laptop and the desktop, without the use of internet access or secondary storage. Acceptable responses were a switch, a hub, a crossover cable or a Bluetooth device. Common erroneous responses included flash devices (which are secondary storage devices) and internal bus protocols such as firewire.

Part (b) asked candidates to describe the functions of the three layers at the bottom of the OSI model. Approximately 90 per cent of the responses correctly state the network, data link and physical layers as the three lower layers of the networking model. The expected descriptions are respectively, the layer that allows routing between networks, the layer that provides reliable data transfer, and the layer that transports bits over physical media for the network, data link and physical layers. Approximately 10 per cent of responses had all descriptions accurate, but most of the remainder were vague. A few responses mentioned the three upper layers of the OSI model.

Part (c) tested candidates' knowledge of wireless networking protocols by asking for a justified statement of whether 802.11a or 802.11b suffered from more interference. Most candidates accurately stated that 802.11b suffered more interference. A few were, however, of the opinion that 802.11b was a more advanced version of 802.11a, and that 802.11a suffered more interference.

Part (d) required candidates to describe spooling with the aid of an appropriate example. Most candidates gave a good example in the form of print spooling. The description of spooling as a strategy that uses temporary storage to interface processes with a high output rate to processes with a lower input rate was however provided by less than a third of the candidates. More than half the candidates provided responses that received partial credit. Some candidates missed the point, however, by indicating that spooling is a strategy for opening files in applications, or a strategy for sharing resources.

In Part (e), candidates were expected to describe the process of transmission of a message between two specified machines on a bus network. Only a few candidates provided the expected response that the transmitting machine broadcasts the message, all machines on the bus receive the message but only the intended recipient performs additional processing, while the other machines discard the message. Most candidates provided a response that earned partial credit. Some candidates suggested that a message could be sent directly from the source to the destination without affecting any other machine.

Part (f) required candidates to state two disadvantages of a bus network topology. Over 50 per cent of candidates scored at least one mark for any of the responses that included low security, single point of failure, limited cable length and reduced efficiency with increasing devices. A common response was that a bus network requires terminators, but that is a feature of the topology, not a disadvantage. Other common responses for which credit was not given included the topology being slow, and assertions that if one node is damaged, the entire network fails.

Paper 03 – School-Based Assessment (SBA)

General performance was reflected in a mean score of 44 out of 60, with a standard deviation of 9.06 per cent and scores ranging from two to 60.

C A R I B B E A N   E X A M I N A T I O N S   C O U N C I L

REPORT ON CANDIDATES' WORK IN THE CARIBBEAN ADVANCED PROFICIENCY EXAMINATION®

MAY/JUNE 2019

COMPUTER SCIENCE

GENERAL COMMENTS

This was the eleventh year in which the revised syllabus was examined. In both Units 1 and 2, there were three examination papers, namely, Paper 01, Paper 02 and Paper 03. In each case, Papers 01 and 02 were examined externally by CXC while Paper 03, the School-Based Assessment, was examined by teachers and moderated by CXC.

In each unit, Paper 01 consisted of multiple-choice questions that were designed to test candidates' breadth of coverage of the syllabus. Paper 02 consisted of essay-type questions that were designed to test candidates' depth of understanding of the syllabus. Thus, candidates were expected to show deeper insight and understanding of the topics examined in Paper 02.

DETAILED COMMENTS

UNIT 1 — FUNDAMENTALS OF COMPUTER SCIENCE

Paper 02 — Essay Questions

Section A — Computer Architecture and Organization

Question 1

Parts (a) (i) and (a) (ii) required candidates to construct a truth table for a given circuit and to write a logic expression for the same circuit. Most candidates completed this task correctly. For Part (a) (iii), candidates were asked to draw a circuit representing the logical expression (NOT X) or (NOT Y) given to inputs *x* and *y*. Approximately 50 per cent of candidates were able to fully complete this task.

Part (b) required candidates to describe the functions of a multiplexor and a decoder. Most candidates were able to describe the functions of the multiplexor but they struggled with the decoder.

For Part (c), candidates were asked to describe with the aid of a diagram, a 2–4 line decoder that is used to manage four gates from a central control. Some candidates did not do well on this task and some of them did not attempt it.

Part (d) asked candidates to describe the usefulness of the program counter register while executing a series of steps. Most candidates misinterpreted the question and gave inappropriate answers.

The question was worth 25 marks. Generally, candidates did fairly well with the majority gaining in the range of 11–15 marks.

Question 2

Part (a) which tested candidates' knowledge of two's complement was divided into three parts. Part (a) (i) required candidates to provide a reason for the suitability of using two's complement to represent integers. Part (a) (ii) asked candidates to state the range of numbers that can be represented by using two's complement format with 16 bits and Part (a) (iii) required candidates to calculate the two's complement representation for a given decimal value. These were completed by most candidates.

Part (b) asked candidates to show a binary number which was in floating point format, as a decimal. Candidates were required to show all of their working. Most candidates had difficulty completing this task.

For Part (c) (i), candidates were asked to state the functions of the MAR, MDR and IR. Some candidates did this task well but many others struggled with the explanation. For Part (c) (ii), candidates were asked to describe the main stages of the instruction cycle; this was completed correctly by most candidates.

Part (d) asked candidates to explain how cache memory could speed up a computer and to give a situation when cache memory would not increase the speed of the computer. Many candidates did not seem to understand the question and therefore gave inappropriate answers.

From observing the results, this question appeared to be difficult for candidates. Out of a maximum of 25 marks, the majority of them scored in the 0–5 range. It is recommended that teachers reinforce number systems and that the topics of registers and memory be emphasized.

Section B — Problem Solving With Computers

Question 3

Candidates were asked to

- list three properties of an algorithm
- write an algorithm to print out multiples of 7
- draw a flow chart from a given algorithm
- write the values to be outputted by the given algorithm based on various input values
- describe any two stages that should be followed when creating software for a construction company which sells equipment and requires software to manage its information.

Some candidates were able to correctly complete the initial tasks correctly; however, generally, the question was poorly done as about half of the candidates who wrote the paper obtained five or less marks. This shows a weakness in the area of problem solving and algorithms.

Question 4

For Part (a), candidates were asked to define the terms *selection structure*, *iteration structure* and *function*. The majority of candidates were able to define selection structure and iteration structure fairly well but they had difficulty defining the term *function*. Part (b) required candidates to distinguish between *bounded iteration* and *unbounded iteration*. Candidates' responses showed that they were familiar with the concepts but they were unable to express themselves fully.

Candidates were given a print function written in C and for Part (c) (i), they were to draw the expected output from the function. Candidates were able to provide correctly drawn patterns. Some did not attempt the task.

For Part (c) (ii), candidates had to rewrite the print function using *for* statements instead of *while* statements. Candidates had difficulty converting from the *while* loops to *for* loops; this shows a weakness in the area of programming.

In Part (d), candidates were asked to write a function that accepts an integer value and returns the factorial of that integer. Some candidates were able to complete this task correctly but many others found it difficult to complete and some did not do it at all.

This question which was worth 25 marks, was not well done; most candidates received marks in the 0–5 range. The areas of problem solving and programming continue to be weak areas among candidates. It is recommended that the school allow teachers with particular strengths to teach particular modules.

Section C — Programming

Question 5

Part (a) (i) required candidates to define the term *debugging* and Part (a) (ii) asked them to state how indentation, comments and descriptive variables can be used to create maintainable code. Most candidates responded accurately.

In Part (b), candidates were given the task of writing a set of C program statements that allow a user to enter timings for a traffic light until all entered values are valid. Candidates were given validity rules for entered timings. The programme candidates wrote was expected to have the following results:

- If timings are valid, output "Timings are valid" and exit
- If timings are not valid, output "Invalid timings: Please re-enter", and accept a new set of values

Many candidates had problems completing this task. Some of them did part of it and some of them did not attempt it at all. Again candidates' weaknesses in the area of programming came through.

In Part (c), candidates were given a piece of code which contained syntax errors and were asked to identify and correct them. Most candidates completed this task correctly; scoring in the 16–20 range, out of a maximum of 25 marks.

Question 6

Candidates were asked in Part (a) to describe the characteristics of the programming paradigms: declarative, procedural and scripting. Many candidates had difficulty completing this task correctly.

Part (b) asked candidates to state two reasons why using modular code is considered good programming practice and Part (c) (i) asked them to state what is a function prototype in C. These parts were well done by most candidates.

For Part (c) (ii), candidates were required to write a function named calcTotal that accepts two integer arguments, and returns the sum of the first argument added to a 20% increase of the second. In other words, calcTotal accepts first and second, and returns first + 1.2 * second. Many candidates had difficulty completing this task correctly.

In Part (c) (iii), candidates were given the following scenario from which to write a C function.

> A local court system is implementing a file access program for civil cases. The objective of the system is to allow the recording of specific information for reporting on a summary of recorded information. Four items of information per case are to be recorded, namely the name of the complainant, the loss suffered, the name of the accused and the court fees. Court fees attract a 20% tax. All names consist of a first name and a last name. The final line of the file contains a hashtag for the complainant's name. A sample of the file is shown below. Note that if the case is thrown out of court, the loss and fees are both set to zero.

John Public	200	Jill Yardly	20
Alice Wonderland	50	Bob Sky	10
Trouble Maker	0	Innocent One	0
#			

Candidates were required to use C code to write a function that loops through the file and reads the loss suffered and court fees for each record, then prints the total cost payable by each of the accused persons. The function calcTotal had to be part of the solution.

Many candidates had difficulty performing this task and some did not attempt it at all. This question was poorly done. A vast majority of candidates earned marks in the 0–5 range; 25 was the maximum mark.

This year, candidates did not seem to fully grasp the programming concepts and it showed as a trend in several questions where programming components were involved. There needs to be some strengthening of the delivery of the programming modules at schools.

Paper 03A — School-Based Assessment (SBA)

Problem Definitions

Problem definitions were not well done in some samples. The provision of a brief context and clear ideas of how the problems were manifested in the organization, along with supporting evidence, were often not shown. Many students provided extensive backgrounds of the organization instead of the problem description. Most students did not pay attention to the requirements when they were writing their problem definition.

Narrative and Flowcharts or Pseudocode

The narrative description of the algorithms was not well interpreted by some students. Narratives were supposed to describe what was designed in the flowchart or pseudocode algorithm. Some algorithms were not properly designed. Students seemed to know the structures, but some were unable to use them to produce good flowchart and pseudocode algorithms. Students need to pay attention to designing flowchart algorithms correctly. Teachers should give students more opportunity to practice in this area.

The majority of students attempted either flowchart or pseudocode algorithms for the SBA, however, some of the programs designed did not match the algorithms. Students were asked to write programs using procedural C only. Some students chose to use such languages as JAVA, PASCAL, C++ or Visual Basic. This is deviating from the aim of the syllabus. It is imperative that teachers and students pay close attention to the syllabus to avoid being penalized during moderation. Students are also advised to print code directly from the compiler and not from a word processor.

UNIT 2 — FURTHER TOPICS IN COMPUTER SCIENCE

Paper 02 — Essay Questions

Section A — Data Structures

Question 1

Part (a) required candidates to briefly describe the concept of an *abstract data type* (ADT). Candidates seemed to have no difficulty providing a brief description.

In Part (b), candidates were given a linked list with four nodes containing the letters H, E, A, P, ordered to spell the word HEAP. Candidates were then asked in Part (b) (i), with the aid of a diagram, to explain how a new node, containing the letter C, can be inserted at the start of the list. In Part (b) (ii), candidates were asked how the node with the letter A can be deleted from the list. Most candidates were able to complete this task correctly but some had a bit of difficulty.

For Part (c), candidates were asked to write C code for the stack operations PUSH and POP. The details given regarding the stack were: the stack is implemented as a fixed size array where TOP and MAX are global variables used to represent the index of the item at the top of the stack and the maximum number of items in the stack respectively. Candidates were expected to represent the code as functions such that the stack array, STACK, is accepted as a parameter. Candidates had difficulty with this task especially with the POP function; this is a weakness in the area of programming.

Overall, this question was fairly well done. The question was worth 25 marks and most candidates received marks in the 16–20 range.

<u>Question 2</u>

In Part (a), candidates were given an array of numbers (34, 56, 12, 89, 2) and had to

- write C code to declare the array
- explain how a selection sort works
- show diagrammatically each pass of the selection sort as it sorted the array.

Some candidates performed these three tasks fairly well but many had problems coding the bubble sort. Some did not attempt it at all. Overall, out of a maximum of 25 marks, performance from the majority of candidates fell within the 6–10 mark range. The common thread running through candidates' responses was weakness in the area of programming.

Section B — Software Engineering

<u>Question 3</u>

Most candidates were able to complete Parts (a), (b) and (c) correctly. Those parts had required that candidates

- list four properties of a well-engineered software product
- briefly outline the main difference between the waterfall approach and evolutionary development
- define what is meant by the term *CASE tools* and support their answers with two examples of how the tools are used.

For Part (d), candidates were given a scenario of a smart refrigerator that uses barcodes on items to facilitate identification, constant stock updating, report generation and the automated processing of payments. Candidates were asked to draw a data flow diagram to represent the scenario. Some candidates had difficulty completing this task correctly. Some had wrong shapes, wrong arrows, wrong entities, wrong processes and no data store.

Overall, most candidates were able, from a maximum score of 25, to receive marks in the 16–20 range.

<u>Question 4</u>

Part (a) required candidates to define the terms *functional requirements* and *non-functional requirements* and provide one example of each to support their answers. Most candidates completed this task correctly.

For Part (b), candidates were given the scenario of an employee database and were asked to state what description, attribute type, length and range elements should be stored for the Employee ID attribute in the data dictionary. This was well done by most candidates. However, candidates had great difficulty explaining the terms *alpha testing* and *beta testing*, which they were asked in Part (c), to define.

In Part (d), candidates were given a scenario of a university with several unique departments each offering courses with distinct course names. Each course was taught by a single professor and one or more teaching assistants. Candidates had to draw an E-R diagram to model the data. Some candidates did well but others had problems like plural entities, wrong shapes, wrong cardinalates and wrong relationships. Most candidates obtained marks in the 16–20 mark range.

Section C — Operating Systems and Computer Networks

<u>Question 5</u>

Part (a) asked candidates to outline the difference between *preemptive scheduling* and *non-preemptive scheduling* algorithms. Part (b) required candidates to state the activities that occur when a process moves through the states: ready to running, running to ready, running to block and block to ready. For the tasks in Parts (a) and (b), candidates seemed unfamiliar with the concepts and therefore gave inappropriate answers.

For Part (c), candidates were told that a process, P5, has been interrupted by another process with a higher priority, P2. Candidates were required to briefly describe the sequence of steps that the operating system would execute to allow P2 and P5 to run to completion. It was noted that there was only one processor. Many candidates were unable to describe the steps.

In Part (d), candidates had to use a clearly labelled diagram to explain how deadlock could occur in a scenario where two processes, P0 and P1, each require the use of two resources R0 and R1. Many candidates gave a definition of what deadlock was but did not use the given processes to explain how deadlock would occur. Some drew incorrect diagrams.

Part (e) required candidates to state two advantages and two disadvantages of the command line over a graphical user interface (GUI). For Part (f), candidates had to describe three techniques one colleague could use to ensure that some sensitive files she needed to share with another colleague could be sent and received securely. Responses to both Parts (e) and (f) were satisfactory.

Overall for this question, candidates' scores fell within the 16–20 marks range.

Question 6

For Part (a), candidates were required to describe the jobs of the network, physical and application layers in the OSI reference model. In their responses, they were expected to give an example of the network devices or protocols which operate at each layer. Candidates had difficulty describing the application layer.

Part (b) asked candidates to list four characteristics of a good password; candidates seemed to have no difficulty responding to this question. For Part (c), candidates had to describe one technique that can be used to fix signal attenuation problems on a network. Candidates were not familiar with the concept and therefore gave inappropriate answers.

Part (d) (i) required candidates to consider a small business which is seeking to set up a network with five desktop computers, a network printer, several files which need to be shared, and access to the Internet. Using this information, candidates had to draw a diagram to illustrate how the network could be constructed so that all computers would be able to access the printer, shared files and the Internet. Most candidates completed this task correctly.

Part (d) (ii) asked candidates to briefly describe how printing, file sharing and Internet accessibility would be possible based on the diagram they drew in Part (d) (i). Many candidates did not seem to understand the question and gave general descriptions on how these things could be done but they did not base their responses on the diagrams they had drawn.

For Part (e), candidates had to order the mobile network technologies LTE, CDMA, GPRS, GSM in terms of speed, from the slowest to fastest. Many candidates got the order incorrect.

Part (f) asked candidates to state two advantages of a centralized network over a decentralized network. Candidates seemed unfamiliar with the concepts and therefore gave inappropriate responses.

Generally, the majority of candidates achieved scores in the 16–20 mark range.

Paper 03A — School-Based Assessment (SBA)

In general, performance throughout was good. However, there were a few students who used the old syllabus instead of the one prescribed by CXC effective from May/June 2009 examination. Students were required to write programs using procedural C only. A few candidates chose to use languages such as JAVA, PASCAL, C++ or Visual Basic. This is deviating from the aim of the syllabus. It is imperative that teachers and students pay close attention to the syllabus prescribed in order to achieve its goal.

Problem Definitions

Some students focused on providing background information and description of the organization instead of concentrating on the requirements (See syllabus Page 31).

Techniques of Analysis

Most students were able to name the techniques of data collection and describe how each was performed. However, some failed to give relevant ones. Incorrect symbols were often used within the data flow and ER diagrams. As a result, students were unable to produce relevant diagrams. In a few cases, the diagrams did not correctly represent solutions to the problems identified.

Functional and Non-Functional Requirements

Most students were able to correctly identify functional and non-functional requirements of the system. However, a few students used hardware and software requirements, for example, ‘processor speeds’ or ‘operating systems’ for this section; this was incorrect.

System Structuring

Most students produced a system structure but failed to give ones that were relevant to the project they pursued.

User Interface Design

Most interfaces were relevant. Many students were able to correctly state the type of interface they would implement but rarely stated the appropriate justification for its use.

Algorithm Design

Some algorithms were not properly designed. Some students seemed to know the structures but some were unable to use them to produce good flowchart and pseudocode algorithms. Students need to pay attention to correctly designing flowchart algorithms. Use of symbols should be practised.

Coding Some Used Languages Other Than C

Some samples were submitted without printed programming code and/or screen shots to verify program functionality. Instead, soft copies were submitted. Generally, this part was well done. Students are advised to print code from the compiler directly and not from a word processor.

Testing

Testing usually focused on normal data and tended not to test abnormal or extreme cases.

Recommendations

Each school should review the syllabus in order to assess and detail the concerns and challenges with particular topics. Schools should also network with each other to optimize the teaching–learning experience.

From observing the responses, the topics of algorithms and programming seem to be the weakest areas. It is recommended that teachers who are good at algorithms and programming be used to teach that module even if they have to be shared by more than one school. Resource persons who have strong programming skills should also be invited to have sessions with the teachers to sure up their skills in these two areas.