**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.

Question 2

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**

Question 3

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.

Question 6

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**

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 identify a problem in the problem statement and therefore full marks were not awarded.

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, 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.

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 is incorrect.

ER Diagram
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.

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 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.