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.

Question 4

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**

Question 5

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.