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.*

SPECIFIC OBJECTIVES

Students should be able to:

4. *describe the main characteristics of a processor.*

CONTENT

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.

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).*

UNIT 1
MODULE 2: *PROBLEM-SOLVING WITH COMPUTERS (cont'd)*

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.*

<u>**Suggested Teaching and Learning Activities**</u>

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.*

Students should be able to:

SPECIFIC OBJECTIVES	CONTENT
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.*

<u>**Suggested Teaching and Learning Activities**</u>

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.*	

UNIT 2
MODULE 2: *SOFTWARE ENGINEERING* (cont'd)

<u>Suggested Teaching and Learning Activities</u>

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.*

<u>Suggested Teaching and Learning Activity</u>

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.

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.

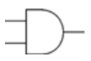
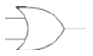
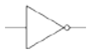
Resit candidates must be entered through a school, approved educational institution, or the Local Registrar's Office.

◆ 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.

<u>WORD</u>	<u>DEFINITION/MEANING</u>	<u>NOTES</u>
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	

<u>Western Zone Office</u>
2008/06/02