REPUBLIQUE DU CAMEROUN
Paix – Travail- Patrie

MINISTERE DES ENSEIGNEMENTS SECONDAIRES

INSPECTION GENERALE DES ENSEIGNEMENTS

INSPECTION DE PEDAGOGIE CHARGEE
DE L'ENSEIGNEMENT DE L'INFORMATIQUE

REPUBLIC OF CAMEROON
Peace- Work- Fatherland

MINISTRY OF SECONDARY EDUCATION

INSPECTORATE GENERAL OF EDUCATION

INSPECTORATE OF PEDAGOGY IN CHARGE
OF THE TEACHING OF COMPUTER SCIENCE

Teaching Syllabus and Course Specifications For High School Computer Science

Observe the environnements and choose better study options for a fulfilled life

LOWER SIXTH GENERAL EDUCATION

July 2019

TABLE OF CONTENTS

1.0 GENERAL INFORMATION

1.1 INTRODUCTION

The high school program in Computer Science is designed to equip students with necessary skills and experience to apply computational thinking practices to solve a problem using a computer. With a unique focus on creative problem solving techniques, representation and ethical use of computer, the course offer in students a broad range of competences, individual skills and motivation which are essential for a successful working life.

This syllabus is expected to provide students with the opportunity to gain national qualifications in Computer Science recognized by educational institutions and employers towards obtaining a degree or to further endeavours in the field or employment. Students acquire knowledge and understanding of the academic aspects of Computer Science through theoretical lessons, practical applications and teamwork.

It is strongly recommended that the teaching of this syllabus should adopt the Competence-Based Approach. This is a task-oriented methodology to give students more computer user-time practices to help them gradually bring out solutions to problems. Many hands-on practical tasks have been suggested to enable students to practice what they will learn. It is hoped that teachers will follow the suggestions and provide opportunities for students to explore and discover more in a computing environment.

1.2 COURSE DESCRIPTION

Computer Science (CSC) curriculum for high school is a two-year rigorous, University entry level preparatory course that builds high school students with broad foundations in the field of modern computing. The course is a continuation of Computer Science curriculum delivered at the Secondary School level. It covers a wide range of fundamental topics including data structures and algorithms designs, programming, information systems, computer organization and networks, digital privacy and security, and the societal impacts of computing.

The total learning time for all the modules on the High School Computer Science curriculum within two years is 426 periods with half the time used in the first year. (A period is an average of fifty minutes.) The way that this time is spent will reflect the subject matter of each module. However, a minimum of 8 periods has been attributed to be the total learning time per week in schools. On the module specifications, the total learning time is divided into two components namely theory time and practical time. Students spend more time working on the computer.

The final year of high school requires students to undertake a project. This enables students to build on their academic work and to apply the technologies learned to a real world scenario.

1.3 AIMS

Computer Science curriculum is intended to build in students:

A) Computer literacy with an understanding of fundamental concepts of computers;
B) Positive attitudes towards contemporary development of information, computational and communication technologies;
C) Problem solving skills;
D) Confidence to analyse and apply theory and practical skills in real world situations through case study and project work.

1.4 OBJECTIVES

After completion of this high school Computer Science course, students should be able to:

1. Demonstrate knowledge, critical understanding, skills and attitudes necessary for adapting to a fast changing technological world.
2. Evaluate possible tools, techniques and approaches required to make sound judgments based on the knowledge relevant to the field of Computer Science.
3. Prove the ability to recognize the nature and extent of information needed and the ability to search for it effectively and efficiently.
4. Use established techniques to critically evaluate, interpret and present the concepts, facts, principles and theories relating to the field of Computer Science for the purpose of communication, comprehension, design and prediction.

5. Demonstrate the ability to select and apply appropriate practices and tools based on analysis to propose satisfactory computer-based solutions to simple problems.
6. Communicate information (verbally, electronically or in writing) accurately, reliably and effectively using appropriate documentation techniques and report formats.
7. Deploy appropriate, practices and tools for the specification, design, implementation, evaluate and testing of a computing solution to a simple problem using a range of software and hardware.
8. Demonstrate awareness of the appropriate professional, social, ethical, moral and legal obligations and practices in the field of Computer Science. Students should develop the ability to learn and work both independently and as part of a team committing to professional development and lifelong learning.

1.5 TEACHING AND LEARNING STRATEGY

Students of high school come from a wide variety of backgrounds. Some are mature students wishing to study in part-time or full-time mode. Some have no formal qualifications. Others come straight from secondary school to study and have narrowly missed the entry requirements for Computer Science in high school. Yet others have qualifications and/or experience in different fields and are attempting to move into a new field by studying Computer Science. Some of these students will have highly advanced transferable skills while others will need to develop theirs. Some students will have good IT skills while others new to the field will not. Owing to these diverse backgrounds, it is important within this high school course of study, particularly in year one, to cater for the requirements of all these students and to lay solid foundations for the theoretical and practical contents, and the associated assessment strategies of the modules within the course.

For students to master this subject, teachers must acquire both a mastery of the subject matter and the pedagogical skills that will allow them to appropriately present the material to students. Teachers should believe that all students can succeed. They have a responsibility to help all students learn, especially students with special education needs and they work collaboratively with special education resource teachers, where appropriate, to achieve this goal. It is understood that there must be a match between the computer science skills and knowledge defined for the students and the acquired skills and knowledge of the teachers. At the same time, teachers must have a greater depth of knowledge than that embodied in the topics they are teaching.

The teacher/facilitator should use different methods and techniques in teaching the subject. The computer science curriculum aims essentially to develop learners' practical skills. To this end, the teacher should focus on practical exercises that builds on theoretical concepts. Several teaching and learning activities have been suggested to the teacher, however, the teacher is free to use other methods and techniques taking into consideration the availability of teaching aids and the students' level and abilities.

It is a recommendation that the teacher should use learner-centred methods or Competence-Based Approach to develop learners' skills such as: Communication, Research, Problem solving, Observation, Creative, Innovative and Social skills. The methods and techniques may be delivered through work groups, individual homework, discussion groups, practical exercises, individual or cooperative research, observation on field trips, case study, forums and collaborative platforms.

1.6 COMPUTER SCIENCE PROJECT

The purpose and goals of project-based learning is to bring out in students:

- A knowledgeable, competent person;
- A critical thinker;
- An effective communicator.

Students are not just consumers of technology but also creators. Computer Science curriculum exploits project-based learning in order to better engage students in the practical processes. By encouraging students to use critical thinking skills and challenging them to solve real life and meaningful problems, project-based learning helps students to develop a deeper and more profound understanding of the power of computation as well as effectively broaden their participation in computing in everyday lives. It is a strategy to help students improve their retention of learned experiences and develop stronger problem solving, critical thinking, and communication skills.

In the final year of high school, every student is required to present a report of a project undertaken. The students are expected to go through the project lifecycle of conception, planning, execution, presentation and evaluation within the two years of the high school program. Shorter projects and practical course work may be based on any parts of the syllabus, but must be structured and presented in a way that meets the objectives of the School-Based Assessment. These mini projects may take 3 to 4 weeks though instructors may tailor durations to suit their pedagogic options. Projects and practical work are seen as part of the student's normal schooling activities, and so these durations are effectively turn-around time, not the effective time spent on the project.

Teachers who are unfamiliar with the goals, methods, and techniques of project-based learning can build their capacity from worksheets and at workshops organized by the different departments of Computer Science in the Ministry of Secondary Education.

1.7 ASSESSMENT AND EVALUATION

The primary purpose of assessment and evaluation is to improve student learning. Information gathered through assessment help teachers to determine students' strengths and weaknesses in the achievement of the curriculum expectations in each module. Students' performance also serves to guide teachers in adapting curriculum and instructional approaches, and in assessing classroom practices and the overall effectiveness of the course.

The nature of this subject area aims to specify that students completing the high school course successfully should be able to apply the skills acquired. Assessment is predominantly practical. However, evaluation throughout the course should be balanced taking cognizance of both theoretical knowledge and practical skills. This course uses three types of assessments:

A. Formative assessment to check for understanding after teaching each sub-topic, topic or module in the form of assignments and quizzes; multiple choices questions, fill-in-the-blank, simple test questions or assignments in the form of roving conference.
B. Summative examination at the end of a sequence, a term, a year or end of course evaluation to find out if the objectives set are attained. These examinations are multiple choice questions, essay questions, case study questions and practical examinations to assess students' ability in applying the concepts covered in the course.

C. School-Based Assessment of Computer Science project at the end of the high school course which challenges students to research, investigate, and solve problems. The project is assessed based on submitted formal project reports in form of a presentation or demonstration, which provides teachers an opportunity to appreciate the extent of a student’s knowledge and understanding of the subject matter including the extent of their written communication skills and their ability to reflect on what they have learnt.

As part of assessment, teachers are required to provide students with descriptive feedback that guides their efforts towards improvement. Teachers are advised to focus on setting practical tests that relate to scientific and daily life situations in order to test students’ reasoning and technical skills.

1.7.1 Achievement Chart

The achievement chart that follows identifies the grading of knowledge and skills in Computer Science. The achievement chart is a standard guide to be used by teachers to make judgments about students’ work.

Assessment criteria	**Total weight**
Theory written exam	50%
MCQ	20%
Practical lab exam	20%
School-based project assessment	10%

1.7.2 The Grading Policy

Bearing in mind that the subject Computer Science is intended to give students opportunities to play around with computers and gain experience in solving problems, teachers will be well aware that written examination may not be the most suitable means of assessment. For this subject, it is recommended that continuous assessment be used and grades, instead of marks, awarded.

The five-point scale that follows provides a record of the learning skills demonstrated by the student in every module of the course through independent work, teamwork, organization, work habits, and initiative.

Remark	Honour Roll	Excellent	Good	Satisfactory	Needs improvement	Failed	Ungraded
% Score	90 and above	80 to 89	70 to 79	60 to 69	50 to 59	40 to 49	< 40
Grade	A	B	C	D	E	F	U

1.8 REQUIREMENTS

1.8.1 Pre-requisite

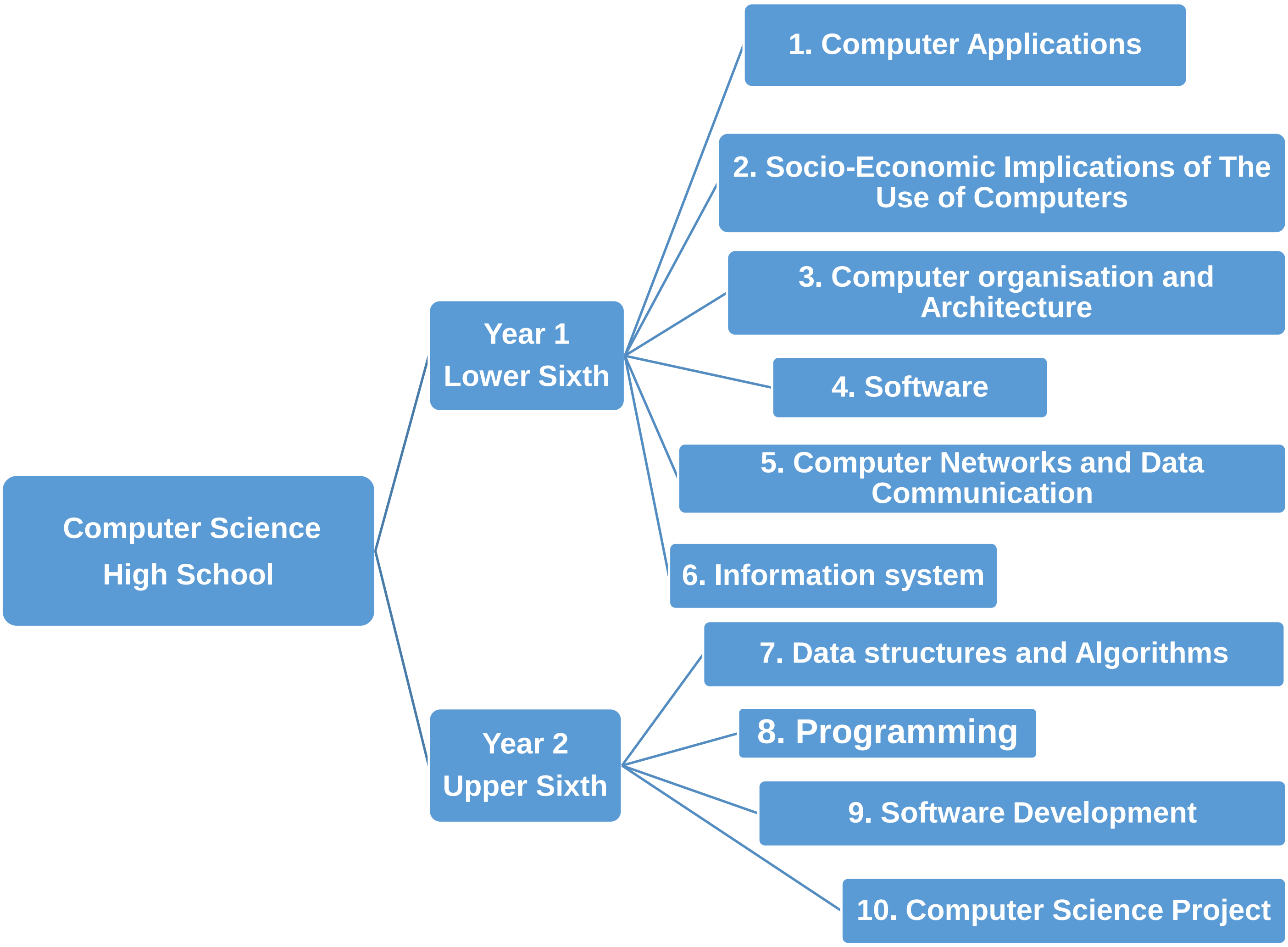
- Students willing to study this high school course require no specific or formal qualification but should have basic computer literacy skills, have understanding for skills acquisition, civic education for general education and responsible social behaviour.
- Students with a working knowledge equivalence to ordinary level Mathematics and Computer Science will find the subject easier than those without.
- Students should have a mastery of Ordinary Level English Language.

1.8.2 Didactic Materials/Equipment

The required materials/equipment to teach this course are the following:

- The availability of one or more computer laboratories. We recommend a ratio of 2 pupils per computer;
- The availability and obtainability of a current and reliable textbook and laboratory manual;
- Typical classroom and laboratory supplies;
- The availability of relevant software specified as didactic materials;
- Internet connection for documentation and research;
- Links to online resources within the content’s learning environment;
- Schemes of work and lesson notes;
- In programming, the C and Pascal programming language should be used as necessary.
- In the case of web and database designs, HTML 5 and CSS or newer versions, Java Scripting language and PHP can be used.
- Office automation skill

1.9 STRUCTURE OF THE COMPUTER SCIENCE CURRICULUM



Computer Science High School

Year 1 Lower Sixth

1. Computer Applications
2. Socio-Economic Implications of The Use of Computers
3. Computer organisation and Architecture
4. Software
5. Computer Networks and Data Communication
6. Information system

Year 2 Upper Sixth

7. Data structures and Algorithms
8. Programming
9. Software Development
10. Computer Science Project

2.0 SCOPE AND SEQUENCE OF YEAR I (LOWER SIXTH) SYLLABUS

Module 2.1: COMPUTER APPLICATIONS			
Class: Lower Sixth	Theory:	Practical:	Duration of Period: 50 minutes

Specific objectives: On completion of this module, students should have the opportunity to:

1. Describe ways in which computing enables innovation.
2. Explain some areas or domain that the computer could be used to facilitate work.
3. List examples of input and output devices.
4. State clearly the various stages of information processing cycle and give examples.
5. Know the various software that can be used in data processing to accomplish tasks.

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCE			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
Computer Applications	• Describe how the computer functions • Describe different types of computers • Explain how they are used and/or what they are used for • Communicate • Collaborate • Share resources	2.1.1 Use of a Computer system	• Identify types of computer devices; • Manipulating computer devices;	• Input devices • Processing devices • Output devices • Storage devices		Printer, camera, keyboard, mouse, processor, chips, smartphones, scanner, monitor, video projector, hard disk, USB key, etc.	2
		2.1.2 Share digital resources	• Cite examples of communication and collaboration tools • Outline the advantages and disadvantages of manual and automated systems ; • Use communication and collaboration tools (PC, Smart Phones, digital cameras, Etc). • Explain/demonstrate how such systems function • Share files on network • Send/Receive emails • Use of social media platforms (FaceBook, WhatsApp, YouTube, etc)	Communication tools collaboration tools	-	• Billing machines • Digital cameras • Smart phones • PCs • Computer • Internet • Mobile or Fixed phones Etc.	4

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCE			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
		2.1.3 Use General purpose and computing applications	- Cite domains of general purpose and computing applications - Identify specific usage in commerce, industry, science, education, arts and media - Use productivity tools (spreadsheets, word processors, database, presentation software, desktop publishing) - Carry out visits to: ✓ Iig Shops ✓ Pharmacies ✓ Banks/IT firms ✓ Insurance Companies	monitoring and control system simulation and modelling systems batch and online processing systems	- Sharing files on a network - Electronic mailing - Retrieving files on a computer - Using the telephone - Basic database design	- Internet - Websites - Architectural/ building plans - Photoshop - Digital camera - ATM machines - Application software	**4**

Module 2.2: ASSESSING THE SOCIO-ECONOMIC IMPLICATIONS OF THE USE OF COMPUTERS			
Class: Lower Sixth	Theory:	Practical:	Duration of Period: 50 minutes

Specific objectives: On completion of this module, students should be able to:

1. Discuss the ways in which innovations enabled by computing affect communication and problem solving.
2. Analyze how social and economic values influence the design and development of computing innovations.
3. Discuss issues of equity, access and power in the context of computing resources.
4. Communicate the legal and ethical concerns raised by computational innovations.
5. Discuss privacy and security concerns related to computational innovations.
6. Explain positive and negative effects of technological innovation on human culture.

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCES			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
Assessing the Socio-Economic Implications of the use of computers	• Cite social challenges associated with the use of computers • Suggest possible remedies	2.2.1 Social and cultural influence	• State Socio-economic effects of computers on people and organisations. • outline measures to combat challenges	• Notion of computer crimes and how to prevent them. • Effects of the use of computer equipment (health, work place, work environment)	• Responsible use of computer • Good ergonomic practices. •	• Charts • Cardboard • Scheme of work • Endorsed textbooks and digital resources	2
	• Discuss the legal issues related to the use of computers : • Copyright laws • Data Protection laws • Privacy • Intellectual property Cyber security	2.2.2 Legislation And Ethical Issues	• Identify laws that prohibit computer crimes,	• Copyright laws stipulations • Knowledge of crimes linked to computer usage • Knowledge of how to recognize licensed open source software	• Awareness of binding legislation of copyright • Awareness of measures to combat computer crimes	• Charts • Cardboard • Syllabus • Scheme of work • Learner guide • Endorsed textbooks and digital resources • Teacher support websites • Discussion forum • Resource List	2
	• Identify and discuss the ethical and moral obligations of the users • Equity, access and power	2.2.3 Economic values	• Recognize privacy and integrity of commercially sensitive or personal material on the internet.	• Knowledge of Professional Standards of conducts, eg. BCS, IEEE, ACM, etc. •	• Awareness of professional, ethical and moral obligations of users of information systems	• Charts • Cardboard • Syllabus • Scheme of work • Learner guide • Endorsed textbooks and digital resources • Teacher support websites • Discussion forum • Resource List	**2**

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCES			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
	• Analyse the economic aspects related to the use of computers • Fostering innovation • Exploitation of information • Limits on information access		• The recognition of economic advantages and disadvantages of computers to society.	• Benefits and limitations of the use of computers in economic domain.	• Awareness of economic effects of the computers on people and organisations.	• Charts • Cardboard • Endorsed textbooks and digital resources • Teacher support websites • Discussion forum • Resource List	**2**

Module 2.3: SOFTWARE			
Class: Lower Sixth	Theory:	Practical:	Duration of Period : 50 minutes

Specific objectives: On completion of this module, students should have the opportunity to:

1. Demonstrate an understanding of software Requirements.
2. Differentiate between application and system software.
3. Know the different types of operating systems.
4. Understand the structure, functions, and philosophy of operating systems.
5. Understand scheduling, dispatch and deadlocks simulation computing.

Contextualization		Competencies		Resources			
Family of life situations	Example of life situations	Category of actions	Examples of actions	Basic (core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
EXPLORING SOFTWARE	- software acquisition - software selection - choose between purchasing an existing software or creating a new one - software installation	2.3.1 Make decisions on suitable software in a given situation	- Define software. - State attributes of a good software - Explain the various methods of acquiring a software: - freeware, open source, proprietary and shareware, … - Distinguish between Open and Closed source software - Compare different software for the same purpose - identify categories of shareware - Distinguish between system software and application software - Know examples of software suitable for each user task. - Know what to do and what not to do with a given software - Acquire and install a software - Uninstall a software	- Notion of software, software types and categories - Types of system software: Operating system, Utility software, - Language translators and Device drivers. - General purpose software e.g MS Office and special purpose software e.g browsers, media players, ereaders, etc - Freeware, shareware and types	- Responsible use of proprietary software	- Textbook - Internet	6

Contextualization		Competencies		Resources			
Family of life situations	Example of life situations	Category of actions	Examples of actions	Basic (core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
	- The use of various user interfaces - Working with OS; - Partition, clean and defragment a hard disk drive - Keeping the computer healthy	2.3.2 Exploring System software	- Elaborate the purpose of System Software - Explain the scope of system software - Stating the main purpose of each type of system software giving examples. - Recognize system software and their classification. - Know the exact role of system software in the computer - Configure of client OS; - Explain concepts such as: Disk partitioning, disk cleaners, defragmentation; - restore or recovery a System; - partition, clean and defragment a hard disk drive; - Install and uninstall an antivirus solution; - Elaborate the need and scope of system software - Listing system software - Classify system software - Define and state examples of compilers, interpreters and assemblers; - Distinguish between an IDE and a text editor - Define and state examples of language translators - Classify operating systems into CL and GUI. - Explain the human computer interface: Command Line Interface (CLI), Menu Driven Interface (MDI) and Graphical User Interface (GUI)	- Role of system software - Operating systems history, evolution and functions. - Utility software and their uses. - Stages of program execution: compiling, linking, - Loading and executing - Types of human computer interface (HCI	-	- Textbook - Internet - Video projector - Computer with an OS.	8

Contextualization		Competencies		Resources			
Family of life situations	Example of life situations	Category of actions	Examples of actions	Basic (core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
	- Use of operating system - Using both CLI and GUI	2.3.3 Operating System and Functions	- Describe the organisation of different OS components to form an OS architecture; - Explain basic functions of OS such as: File Management, Process Management, Memory Management, User interface, Handling of errors, providing security and some others; - Describe the operating system kernel; - Explain types of kernel, shell and libraries; - List some operating systems. - State and explain the components of an OS - Understand the architecture of an operating system. - Explain the role of an operating system - State and explain the functions of an operating system	- Notions on operating system structure and architecture. - Basic functions of the operating system	-	- Textbook - Internet - Video projector - Computer with an OS.	4
	- Monitor processes: process execution and system resources using windows task manager - Selection of an efficient scheduling policy to solve a given scenario.	2.3.4 Process Management	- Definition of a process - State the types of information found in the Process Control Block (PCB) - Describe/explain the Process state and process transition; - Explain what is Process synchronization, cooperating processes, and context switching. - Know basic concepts such as: Race Condition, mutex, and semaphore - Understand processor sharing and types of schedulers; - Know preemptive and non-preemptive scheduling algorithms with examples; - Understand process state transition and reason(s) for each transition.	- Notions on process and process synchronization. - Categories of schedulers - Processor sharing and scheduling algorithms - Inter-process communication	-	- Textbook - Internet - Charts - Cardboard	10

Contextualization		Competencies		Resources			
Family of life situations	Example of life situations	Category of actions	Examples of actions	Basic (core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
			• Explain how process synchronization enables multiprocessing. • Explain process preemption and context switch • Explain the different categories of schedulers such as: long term (or Job scheduler), Middle term and short term (or CPU scheduler or Dispatcher) • Explain scheduling policies such as: FCFS, SJN, SRT and RR				
	• Utilization of computer memory; • Management of programs in the memory	2.3.5 Memory management	• Explain what is MMU • Explain the need for memory management • Explain concepts like external segmentation, paging and segmentation • Know contiguous memory allocation policies such as: first fit, best fit, worst fit etc • Explain the concept of swapping. • Explain the concept of virtual memory, memory address translation, address space protection; • Explain the various activities of the OS with regards to memory management. • Explain external fragmentation and propose solutions: compaction, paging and segmentation. • Distinguish between paging and segmentation • Understand the role of virtual memory in the computer	• Fixed and variable memory allocation • Contiguous memory allocation algorithms • Operating system memory management.	-	• Textbook • Internet • Charts	6

Contextualization		Competencies		Resources			
Family of life situations	Example of life situations	Category of actions	Examples of actions	Basic (core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
	● Selection of hardware devices in case of purchase	2.3.6 Device management	• Distinguish between device controller and device driver. • Define an interrupt and state the various types. • Explain how the OS handles interrupt. • Explain the concept of spooling and buffering • Know various I/O management techniques such as: Program I/O (Polling), Interrupt Driven I/O and DMA. • Understand device controller which is a hardware and a device driver which is a software; • Explain external and internal interrupt. Know what is an exception and types of exceptions (overflow etc) • Understand how the performance can be increase by using SPOOLING; • Differentiate Hardware interrupt from software interrupt	• CPU- I/O communication • Interrupts and interrupt management in OS • SPOOLING AND BUFFERING		• Textbook • Internet • Charts	6
	Access the register and study different configuration files	2.3.7 File management	• Define a file, file characteristics. • Explain what is file format and give example • Distinguish between a file system and a system file. • Explain the various methods of file access; • Create and manage files in the computer; • Explain file system and state examples such as: FAT16, FAT32, NTFS, ext in unix environment. • Explain and differentiate between serial and direct file access methods	• File characteristics such as: filename, file extension, file size, type of file etc are required; • General notions on files. • File access methods • Working with files in a windows environment	-	• Textbook • Internet • Charts • A PC with a windows OS	6

Contextualization		Competencies		Resources			
Family of life situations	Example of life situations	Category of actions	Examples of actions	Basic (core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
	Other Functions of the OS	2.3.8 OS maintenance	• Explain how the operating system offers security • Explain how the OS handles errors; • Explain the basic security measures offered by the OS: password protection, logins and authentication. • Explain how the OS intervenes in error detection, error recovery	• Error management in OS.	-	• Textbook • Internet	4

Module 2.4: COMPUTER ORGANIZATION AND ARCHITECTURE			
Class: **Lower Sixth**	Theory:	Practical:	Duration of Period: 50 minutes

Specific objectives: On completion of this module, students should have the opportunity to:

1. Demonstrate an awareness of the nature of the hardware involved in computer systems.
2. Appreciate the choice of a combination of particular types of peripheral devices, the operating system and the processor.
3. Understand and analyze computer systems architecture.
4. Explain the structure and functioning of computer instruction set.
5. Describe the organization of different bus systems, and their characteristics in a computer system.
6. Understand low-level parallelism and its implementation in a processor.
7. Know basic logic gates.
8. Carry out arithmetic operations with basic digital circuits.

Contextualization		Competencies		Resources			
Family of life situation	**Examples of life situation**	**Categories of actions**	**Examples of actions**	**Basic (Core) Knowledge**	**Attitudes**	**Didactic material**	**Duration(in period(s))**
Computer organization	• Scope of computer organization	2.4.1 Computer environment	• Describe the functional setup of computer system • Identify and explain the role of different internal hardware components	• concept of computer organization • Notion of computer system environment • Control unit; arithmetic and logic unit; registers; bus system; ROM, RAM		• Text books • Chalk • Internet • Charts	6
	• Hardware selection	2.4.2 Hardware	• List the different components and their characteristics • List different types of computers • Different types of processor architecture • Describe processor hardware • Categorizing hardware (Processor, Storage unit/devices, I/O units) • Processor architecture • Determine a required hardware need from its characteristics and interconnection • Selecting hardware and media for specific activities • Determine the required hardware from types of applications	• Hardware • Mainframe, minicomputer … • RISC, CISC, SISD, SIMD, MIMD • Standard buses and interfaces		• Text books • Chalk • Internet • Charts	10
Choice of a Computer with an appropriate architecture	• Identifying different computer architectures	2.4.3 Computer architecture	• Describe the functions of the various components of the Von Neumann stored program concept. • Comparing Von Neumann with other architectures like Flyin, CRAY, Data flow and Harvard,	• Von Neumann; Flyin, CRAY, Data flow and Harvard architectures, • shared memory • Separate memory		• Text books • Chalk • Internet • Charts	12

Contextualization		Competencies		Resources			
Family of life situation	Examples of life situation	Categories of actions	Examples of actions	Basic (Core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
			• Recall the outline of the Von Neumann architecture and contrast with computers without memory. • Differentiate between shared memory and separate memories for data and instructions • General understanding of the Von Neumann stored program concept. • Determine the possible outcomes of the Von Neumann Machine				
	• Describing parallel processing • Exploring machine instructions	2.4.4 Parallel processing and machine instruction Cycle	• Draw the instruction cycle block diagram. • List the various registers involve in each phase • Illustrating pipelining with diagrams • Describe the Fetch-decode-execute cycle • Understand the stages involve in processing a piece of instruction • Recognize a parallel processing system like pipelining, multiprocessing	• Notion of CPU registers • Notion of data processing		• Text books • Charts • Chalk • Internet	8
	• Describe Polling and Interrupts	2.4.5 Polling and Interrupts	• Demonstrate situations in which polling is done and where interrupts occurred • Make a clear distinction between a device which poll and interrupts	• Notion of instruction execution • Polling • Interrupt		• Text books • Chalk • Internet	2

Contextualization		Competencies		Resources			
Family of life situation	Examples of life situation	Categories of actions	Examples of actions	Basic (Core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
	• Describing: • Machine instruction codes • Instruction set • Modes of operand addressing: immediate, direct, indirect, indexed, relative Assembly language principles: instruction format; addressing modes; typical instructions; assemblers and loaders	2.4.6 Low level programming	• Illustrate the use of instruction codes using exercises • Access instructions and data using the various addressing modes • Recognize and differentiate between Operand addressing modes. • Manipulate machine instructions code (LOAD, STORE, ADD)	• Notion of interpreting machine instructions • Notion of machine languages • Transfer, logical, shift, arithmetic, branch, input and output; register instructions • assembly language instructions	• Appreciate machine instructions code	• Text books • Chalk • Internet	10
	• Listing the different base in which numbers can be represented • Computing arithmetic operations in base 2: addition, subtraction, multiplication,	2.4.7 Binary Arithmetic	• Convert from decimal to binary and vice versa • Convert from hexadecimal to binary and vice versa • Convert from octal to binary • Addition in binary	• Notion of data representation • Carry out arithmetic operations (Addition, Subtraction ..) • sign magnitude / sign absolute, one's and two's complement. • Conversion from one base to another	• Working with different bases such as binary, octal, decimal, and hexadecimal	• Text books • Chalk • Charts • Blackboard	10

Contextualization		Competencies		Resources			
Family of life situation	Examples of life situation	Categories of actions	Examples of actions	Basic (Core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
	division and complements						
	• Defining Boolean logic and logic gates • Presenting the different symbols • Explaining Boolean Algebra and Boolean expressions, and draw truth tables • Identifying basic logic functions involving: Negation (NOT), Conjunction (AND), Disjunction (OR), NAND, NOR and XOR.	2.4.8 Boolean arithmetic and logic gates	• Sketch diagrams for different logic gates symbols • Using truth tables to represent expressions • Simplifying Boolean expressions	• Notion of computer architecture • Notion of digital arithmetic	• Recognizing logic circuits from their symbols • Deduce the output of a simple digital circuit • Apply De Morgan's theorem.	• Charts • Text books • Rulers • Chalk • Blackboard • Pencils	12

Contextualization		Competencies		Resources			
Family of life situation	Examples of life situation	Categories of actions	Examples of actions	Basic (Core) Knowledge	Attitudes	Didactic material	Duration(in period(s))
	• Digital circuits e.g. Traffic light	2.4.9 Basic digital circuits	• Draw the basic symbols for combinational circuits and Adders • Produce truth tables for Adders • Deduce output and input from giving circuits • Deduce types of adders from sketch diagrams • Deducing output and inputs of given circuits and circuit components, binary adder and other basic circuits	• Notion of Boolean logic and Logic gates • Notion of Boolean expressions		• Chalk • Charts • Textbooks	8

Module 2.5: COMPUTER NETWORKS AND DATA COMMUNICATION

Class: Lower Sixth	Theory:	Practical:	Duration of Period: 50 minutes

Specific objectives: On completion of this module, students should have the opportunity to:

1. Appreciate the need for data communication networks.
2. Identify the different equipment and components used.
3. Know all transmission modes, and media.
4. Describe the various network communications standards
5. Recognize the need for communication Protocols.
6. Understand modulation and multiplexing.
7. Explain the concepts of computer Networks and Topology.
8. Understand network implementation and security.

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCE			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
Computer network and data communication environments	• Choosing an appropriate network topology • Troubleshooting a network	2.5.1 Exploration of computer network platform	• Identifying the different types of networks • Describing the different physical and logical topologies • Stating the advantages and disadvantages of different network topologies • Troubleshooting a network • Expressing better ways of connecting network devices • Pairing communication devices. • Describe multitasking • Defining and differentiating types of network architectures • State the role of each architecture	• Computer network • Network topology • Network operating system • Network architecture • Peer to peer network • Server computer base network	• Creative thinking	• Network • Video Tutorials	
	• Setting up a network • Choosing appropriate network components	2.5.2 Identification of different equipment and components	• Explain the functioning of communication hardware and transmission media • State the roles of data transmission devices • Selecting network devices from group of computer devices for specific purpose • Identify the Network Interface Card (NIC) and its functions • Describe the roles of the various components of a data communication system • Identifying network devices from group of computer devices • Configuring a computer network • Enumerating the functions of the different Network Hardware such as MODEM, repeaters, switches, bridges, routers, and gateways • Linking of network devices	• Wireless network • MODEM • Switch • Bridge • Repeater • Router • gateways • NIC	-	-	**6**

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCE			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
			• Identifying the way in which network devices are inter related • Stating how wireless communication occurs				
	• Selecting a convenient transmission technology for a given situation	2.5.3 Data Communication	• Describe the different modes of data transmission (simplex, half duplex, full duplex); • Describe the different types of data transmission (parallel, serial (synchronous Vs Asynchronous)) • Identify features of data communication network • Explain the meaning of communication system. • Explain the purpose of data transmission • Explain why the constant evolution in data communication • Describe the role of various networking devices such as multiplexers • Recognize serial, parallel, broadband, base band, and wireless transmission • differentiate broadband from baseband transmission • describe different types of transmission media (guide and unguided media) • give examples of different types of transmission media • describe the different type of signal transmission (analogic, digital) •	• Data transmission • Multiplexer • Parallel data transmission • Serial data transmission • Synchronous transmission • Asynchronous transmission • Baseband transmission • Broadband transmission • Analogic signal transmission • Digital signal transmission	• Ethical and Self- discipline • Creative and Logical reasoning • Communicate effectively	-	**6**
	• Implementation of data protection mechanism	2.5.4 Data protection	• State the need for securing data during transmission • Give binary representation of data	• Parity bits • Hamming code • Check sum • Polling			**8**

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCE			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
	- Protection of data - Prevention of data unavailability - Network protection		- Explain different mechanism of securing data during transmission - Explain the risk of data damage or theft over the network - State the importance of data backup - State quality of good password - State data protection mechanism - Describe different type of error detecting code (parity bits, hamming codes, cyclic redundancy checks/check sum) - Describe computer malware (virus, worm, Trojan, etc) - Give mechanism for fighting against malware - State the difference between interrupt and polling - Describe the concepts of buffer, interrupts, interrupts priority, handshaking and polling - Express the need for two communicating systems to relate - Identify the risk and benefit in the use of networks - Identify the limitations of networking	- Interrupt - Temporary memory - Data backup - Firewall - Data encryption - malware			
	- Need to establish communication rules	2.5.5 Use of standards and protocol	- Define the roles of standards in data transmission - Describe the OSI reference mode - Stating the roles of network protocols and internet protocols - Stating the different protocols in data communication - Stating why communication system needs protocols to communicate	- OSI - Protocol			**4**
	- sending e-mail - Job search	2.5.6 Use of internet	- Give a brief history of the advent of the Internet	- Internet - Intranet			

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCE			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
	• Participation in social and cultural activities • completion of an electronic transaction • choice of an internet service • searching for information on the internet		• Cite the purpose and use of intranet, Extranet and internet • Describe the different ways available to connect to the Internet • Enumerate the different services available on the Internet • List the advantages and disadvantages of using internets in specified fields of life and studies • Identifying and use software used in locating specific resources from the internet • Navigating among sites using the web browser. • Using an e-mail program to send and receive messages • List Advantages and disadvantages of email compared to the postal system • Send and receive emails with or without attachments • State the differences between Internet and world wide web • Use a search engine to Download a digital resource • State the risk associated with downloading digital resources from the internet • Describe different internet service (ecommerce, ebanking, electronic fund transfert, search engine etc)	• Extranet • Internet services			**8**

Module 2.6: INFORMATION SYSTEM

Class: Lower Sixth	Theory:	Practical:	Duration of Period: 50 minutes

Specific objectives: On completion of this module, students should have the opportunity to:

1. Demonstrate knowledge and understanding of main aspects of Information Systems.
2. Understanding the importance of modeling data as sets of entities with common characteristics.
3. Use a data model as consisting of entities and relationships, representing, querying and updating these as a relational data base.
4. Surveying the functions and services of a data base management system.
5. Demonstrate an understanding of the components of an information system and the links between them.
6. Introducing data flow diagrams and their use in the description of an information system.
7. Understand the need for designing user interfaces, and becoming familiar with design principles.
8. Gain experience in designing and setting up a simple data base.

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCE			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
Development of an Information system	• Management of Information system • • Identification of IS components • • Choosing data capture methods	2.6.1 Information System environments	• Give the different component of an IS • Describe the role of each component in an IS • State what makes up an information system • Give the role of each those things that make up an IS • Give the meaning of a system • Give some example of data capture methods • Explain the need of an IS in an organization • Explain the need for a particular component of an IS at any given instant • Identify some areas where information system can be used real life • State the need of some types of IS (HIS, EIS, GIS, Market IS, and Library IS) • Discuss the importance of a given type of IS in real life • Identify some IS in school • Give the difference between data and information • Identify some of the ways an IS can ease the work of a school Principal as an administrator • Describe some data capture methods • Site the importance of an IS to the school Library, etc • Identify some areas in a given organization where there is need for an IS • Describe the roles of people, org, and technology in IS	• Data • Information • System • IS • HIS • EIS • GIS •	-		

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCE			
Family of life situations	**Examples of life situations**	**Categories of Actions**	**Examples of Actions**	**Basic (core) knowledge**	**Attitudes**	**Didactic Material**	**Duration(in period(s))**
			- State the need for an IS at each level of an organization				
	- Modelling systems - Designing a computerised system - Developing Information Systems(IS) - Representing data for a given problem	2.6.2 Information System Design	- Listing the different design methodologies and the stages involved - Describe some SDLC models - State the advantages of a design method of the other - Describe the following relationships in real life: one-to-one, one-many, many-one, and many-many - Carry out basic ER Modelling - Design of simple IS - Describe the various levels of data modelling - Carry out normalization of databases - Identify the different normal forms - Give the meaning of data redundancy - Cite some problems posed by data redundancy - Explain the need for data integrity and consistency in database - design a relational model - Identify properties of 1NF, 2NF, 3NF - Convert from 0NF to 3NF - Describe some DB anomalies	- Entities - Attributes - Relationships - 1NF - 2NF - 3NF - ER model - Data redundancy - Data integrity - Data consistency - V-model, waterfall model,			8
	- Implementation of database design - Building IS - Developing dynamic web applications	2.6.3 Web Applications and Database implementation	- Implementing PHP and HTML codes - Give some importance of DBMS - Working with RDBMS - Writing queries in QBE and SQL - Populate and retrieve data - Writing scripts (PHP, HTML, JavaScript)	- QBE, SQL, DBMS, RDBMS			**12**

CONTEXTUALIZATION		COMPETENCIES TO BE ATTAINED		RESOURCE			
Family of life situations	Examples of life situations	Categories of Actions	Examples of Actions	Basic (core) knowledge	Attitudes	Didactic Material	Duration(in period(s))
			- Publish works in PHP and HTML on the Internet - Generate reports				

3.0 CONCLUSIONS

“Computational thinking will be a fundamental skill used by everyone by the middle of the 21st Century, just like reading, writing and arithmetic.” *Wing (2011) Computational Thinking*.

It is no new news that Computer Science is a mainstream discipline that can no longer be ignored in schools. This model curriculum provides a basis by which vocational schools, high schools of education and individual initiatives can effectively implement a coherent computer science curriculum that is accessible to all. More work has been done to bring the model close to teaching and laboratory materials that are pedagogically viable and widely accessible. We hope students, teachers, and other external professional sources will support this work by providing appropriate incentives that will enable such a curriculum development effort to succeed.

4.0 ACKNOWLEDGMENTS

The conception and design of this curriculum model has been produced with contribution, feedback and advice from many persons and organisations. We would like to thank the following persons and organisations for their valuable inputs and support for the development of this high school Computer Science curriculum:

S/N	Name	Post	Structure
1	Moudji Didier Rene	ICG Info/Supervisor	IP/Info MINESEC
2	Kambiwoa Lawrence	CS-IF/ Moderator	IP/Info MINESEC
3	Agwe Lucas B.	ICR/CMR/Coordinator	RDSE-SW Buea
4	Ngwa Andrew M.	ICR/ CSC Chief Examiner	RDSE-West Baffousam/GCEB
5	Arrey-Njok Tabe Takang	IPR/ICT Lead Examiner	RDSE –SW Buea/GCEB
6	Anyam Ephraim Awa	IPR	RDSE-SW Buea
7	Chifu Emmanual Bawe	IPR/ CSC Ass. Chief Examiner	RDSE –SW Buea/GCEB
8	Dr. Musongong N. Luke	IPR	RDSE –SW Buea/UB
9	Ikome Joseph Njie	IPR/ICT Chief Examiner	RDSE –SW Buea/GCEB
10	Dr. Mrs Patricia Temeching	IPR	RDSE –SW Buea/UB
11	Clive W. Ayumbi	IPR	RDSE –SW Buea/SWATICT
12	Tangmo Celestine Fozao	IPR	RDSE –SW Buea/SWATICT
13	Dr. William SHU	Assessor A Level Computer Science	University of Buea
14	Belinga Alain Roger	IPN INFO	IP/Info – MINESEC
15	SAHA Pierre	IPN INFO	IP/Info - MINESEC

5.0 REFERENCES

1. CS Teachers.org (2017). Computer Science teachers Association; K-12 Computer Science Standards, Revised 2017
2. Kigali, (February 2010). Computer Science Curriculum for Computer Science Economics and Mathematics Option & Mathematics Physics and Computer Science Option, www.ncdc.gov.rw.
3. Massachusetts K-12 (2011), Computer Science Curriculum Guide: Exploring Computer Science version 4, Joanna Goode University of Oregon, Gail Chapman University of California, Los Angeles Computer Science Equity Alliance.
4. UNESCO – Nigeria Project (2004). Computer Science Higher National Diploma Curriculum and Course Specifications, National Board for Technical Education, Federal Republic of Nigeria
5. Pearson Edexcel (2016). GCSE (9 - 1) in Computer Science, Issue 4.
6. Irv Englander, (2009). The Architecture of Computer Hardware, System Software, and Networking: An Information Technology Approach, Fourth Edition; John Wiley & Sons, Inc.
7. https://k12cs.org/wp-content/uploads/2016/09/K–12-Computer-Science-Framework
8. The Ontario Curriculum Grade 10 to 12 (2008), Computer Studies, Revised, www. edu.gov.on.ca.
9. www.pearsonschools.co.uk/GCSE, May 2019
10. https://www.khanacademy.org/resources/teacher-essentials/teaching-computing, May 2019.
11. K–12 Computer Science Framework. (2016). Retrieved from http://www.k12cs.org
12. CSTA, A Model Curriculum for K–12 (October, 2003). Computer Science: Final Report of the ACM K–12 Task Force Curriculum Committee
13. ACM/IEEE-CS Joint Curriculum Task Force. Computing Curricula 2001: Computer Science Volume. December2001.
14. http://www.acm.org/sigcse/cc2001/Task Force of the Pre College Committee of the Education Board of the ACM.
15. ACM model high school Computer Science curriculum. Communications of the ACM, May 1993.
16. AP Course Description: Computer Science. May 2002.
17. International Society for Technology in Education (ISTE), National Educational Technology Standards for Teachers, June 2002. http://www.iste.org/standards.