



ZIMBABWE

MINISTRY OF PRIMARY AND SECONDARY EDUCATION

COMPUTER SCIENCE SYLLABUS [NON FORMAL] LEVEL 3

2015 - 2022

Curriculum Development Unit
P. O. Box MP 133
Mount Pleasant
HARARE

© All Rights Reserved
2015

ACKNOWLEDGEMENTS

The Ministry of Primary and Secondary Education wishes to acknowledge the following for their valued contributions in the development of this syllabus:

- The National Computer Science Panel comprising of representatives from:
 - Computer Science Teachers
 - Zimbabwe School Examinations Council(ZIMSEC)
 - Teachers' and Technical Colleges
 - Universities
 - Computer Society of Zimbabwe (CSZ)
- United Nations International Children's Education Fund (UNICEF)

TABLE OF CONTENTS

| Contents | Page |
|--|------|
| ACKNOWLEDGEMENTS..... | i |
| 1.0 PREAMBLE..... | 1 |
| 1.1 Introduction..... | 1 |
| 1.2 Rationale..... | 1 |
| 1.3 Summary of Content..... | 1 |
| 1.4 Assumptions..... | 1 |
| 1.5 Cross-Cutting themes..... | 2 |
| 2.0 PRESENTATION OF SYLLABUS..... | 2 |
| 3.0 AIMS..... | 2 |
| 4.0 SYLLABUS OBJECTIVES..... | 3 |
| 5.0 METHODOLOGY..... | 3 |
| 6.0 TOPICS..... | 4 |
| 7.0 SCOPE AND SEQUENCE CHART..... | 5 |
| 8.0 COMPETENCY MATRIX..... | 7 |
| 9.0 ASSESSMENT..... | 17 |
| 9.1 Assessment Objectives..... | 17 |
| 9.2 Scheme of Assessment..... | 18 |
| 9.3 Specification Grid..... | 20 |
| 9.4 Paper Descriptions..... | 22 |
| 10.0 GLOSSARY/APPENDICES..... | 23 |
| APPENDIX I: GLOSSARY OF TERMS USED IN QUESTION PAPERS..... | 23 |

| | |
|---|----|
| APPENDIX II: ACRONYMS..... | 24 |
| APPENDIX III: PRACTICAL ASSESSMENT GUIDELINES | 25 |
| APPENDIX IV: RESOURCES AND EQUIPMENT | 30 |
| APPENDIX V: SUGGESTED REFERENCE BOOKS..... | 31 |

1.0 PREAMBLE

1.1 Introduction

The compressed Computer Science syllabus follows the Science, Technology, Engineering and Mathematics (STEM) domain and is designed to cover level 3 of secondary education. This learning area focuses on foundational principles, practices of computation and computational thinking including their application in design and development of computer systems. This syllabus is designed for learners with a background in computer science and mathematics who want to pursue careers in computer science and other technology related fields.

1.2 Rationale

The increased importance of computer-based solutions provides an important economic opportunity for the society. Computer Science enables learners to acquire knowledge and attitudes to enhance usage of computer technologies for all areas of human activities. The syllabus facilitates the promotion and preservation of indigenous knowledge systems, heritage and culture through the design and development of computer systems. The syllabus empowers learners to pursue a career in Computer Science or related disciplines. The course empowers learners to innovatively develop solutions that can be applied to real life situations. It equips learners with problem solving skills, critical thinking, collaboration, innovation, self sustenance, professional development and lifelong learning.

1.3 Summary of Content

This Computer Science syllabus covers theory and practical activities for learners in Level 3. It focuses on system development, networking, data representation, computer architecture, algorithm design, programming and databases as major learning areas.

1.4 Assumptions

It is assumed that learners:

- have passed mathematics, pure sciences and computer science
- are motivated to learn Computer Science

- are conversant with at least one programming language
- can interpret user requirements to produce economically viable solutions

1.5 Cross-Cutting themes

The teaching and learning of Computer Science should integrate the following cross cutting themes:

- Life skills
- Enterprise skills
- financial literacy
- Heritage Studies
- Environmental issues
- Disaster Risk management
- Collaboration

2.0 PRESENTATION OF SYLLABUS

The compressed Computer Science syllabus is presented as one document for level 3.

3.0 AIMS

The syllabus aims to enable learners to:

- 3.1 develop critical thinking skills and strategies required to carry out research in computer science
- 3.2 appreciate the benefit of computer science in solving real problems
- 3.3 develop an understanding of the relationship between different components of computer systems.
- 3.4 acquire knowledge on the range of computer applications, their social and economic implications
- 3.5 apply moral and ethical approaches to the use of ICTs (Unhu/Ubuntu/Vumunhu)
- 3.6 appreciate development trends in the field of computing
- 3.7 appreciate the role of security in computer systems

4.0 SYLLABUS OBJECTIVES

Learners should be able to:

- 4.1 apply programming concepts to solve real life problems
- 4.2 design computer systems based on user requirements specifications
- 4.3 demonstrate an understanding of data representation in computer systems
- 4.4 apply the fundamental principles and concepts of computer science in algorithm design
- 4.5 demonstrate an understanding of computer architecture
- 4.6 apply ethical principles and standards in the field of computing
- 4.7 apply security measures to protect computer systems
- 4.8 show awareness of the data protection legislation and intellectual property rights
- 4.9 analyze technological changes and make informed decisions

5.0 METHODOLOGY

The teaching and learning of Computer Science is based on an inclusive learner centred approach. The following methods are recommended:

- Problem solving
- e-Learning
- Multi-media
- Simulation and modeling
- Discovery
- Experimentation
- Design based learning
- Project-based learning
- Question and answer
- Demonstrations
- Discussion
- Educational Tours
- Research and Presentations

Time Allocation

The learning area should be allocated at least 2 hours per week. Learners should be engaged in at least **one**: Educational Tour, Exhibition, Seminar per year and; a week of practical orientation.

6.0 TOPICS

The syllabus consists of 9 topics as follows:

- 5.1 Data Representation
- 5.2 Networking
- 5.3 Computer Architecture
- 5.4 Security and Ethics
- 5.5 Software Development
- 5.6 Algorithm Design and Data Structures
- 5.7 Programming
- 5.8 Databases
- 5.9 Enterprising

7.0 SCOPE AND SEQUENCE CHART

| TOPIC | LEVEL 3 |
|----------------------------------|--|
| 7.1 Data Representation | <ul style="list-style-type: none">• Number bases• Floating point arithmetic• Computer arithmetic errors• Data representation |
| 7.2 Networking | <ul style="list-style-type: none">• Open Systems Interconnection (OSI) Model• Transmission Control Protocol (TCP)/Internet Protocol(IP) Model• IP Addressing• Domain Name Systems• Routing Protocols• Cloud Services Models• Cloud Types |
| 7.3 Computer Architecture | <ul style="list-style-type: none">• Hardware Architecture• Software Architecture |
| 7.4 Security and Ethics | <ul style="list-style-type: none">• Data privacy and Integrity• Common threats and software vulnerabilities to computer systems• Protection, access control, and authentication• Legislation on computer security and crimes• Disaster Recovery methods• Risk Management techniques• Code of ethics at the workplace• Business ethics• Impact of social media• Security Policies• Laws and Computer Crime• Environmental laws and issues• Ethical principles |

| TOPIC | LEVEL 3 | FORM 6 |
|---|---|--------|
| 7.5 Software Development | <ul style="list-style-type: none"> • Systems Development Life Cycle (SDLC- Waterfall Model) • Object Oriented Methodology • Prototyping | |
| 7.6 Algorithm Design and Data Structures | <ul style="list-style-type: none"> • Pseudocode structures • Standard algorithms • Data structures • Binary trees and array operations • Primitive data types • Recursion | |
| 7.7 Programming | <ul style="list-style-type: none"> • Programming language features • Structured programming • Functions and procedures • Arrays • File handling • Interface design • Object Oriented Programming • Advanced Programming | • |
| 7.8 Databases | <ul style="list-style-type: none"> • File based database systems • Database Management Systems (DBMS) • Relational Database modeling • Database management | |
| 7.9 Enterprising | <ul style="list-style-type: none"> • E-Business <ul style="list-style-type: none"> - E-Commerce • Intellectual Property Rights • Application areas of Computer Science • Business Proposal Development <ul style="list-style-type: none"> • Telecommunications • Business and enterprise | |

8.0 COMPETENCY MATRIX
Level 3

| SKILL/TOPIC | OBJECTIVE Learners should be able to: | CONTENT (KNOWLEDGE, SKILLS, ATTITUDES) | SUGGESTED LEARNING ACTIVITIES AND NOTES | SUGGESTED LEARNING RESOURCES |
|--------------------------------|--|--|---|--|
| 8.1 Data Representation | <ul style="list-style-type: none"> • convert number bases • multiply and divide binary numbers • normalize floating point binary numbers • represent data and character sets • distinguish arithmetic errors • interpret arithmetic errors | <ul style="list-style-type: none"> • Number bases <ul style="list-style-type: none"> - binary - octal - denary - hexadecimal - 2s Compliment • Floating point arithmetic <ul style="list-style-type: none"> - normalization of floating point numbers • Computer arithmetic errors <ul style="list-style-type: none"> - overflow - underflow • Data representation <ul style="list-style-type: none"> - Character sets - ASCII - UNICODE - BCD - EBCDIC | <ul style="list-style-type: none"> • Changing one number base to another • Calculating number bases • Converting floating point numbers • Changing numbers to binary using character sets • Identifying arithmetic errors • Comparing arithmetic errors | <ul style="list-style-type: none"> • Calculators • Print and electronic media • Data set tables • ICT tools • Internet |
| 8.2 Networking | <ul style="list-style-type: none"> • compare OSI and TCP/IP models • explain the format of an IP address • distinguish between public and private IP addresses • explain the role of DNS • describe Routing Information Protocol(RIP) and Open Shortest Path First (OSPF) routing protocols | <ul style="list-style-type: none"> • OSI Model • TCP/IP Model • IP Addressing • Domain Name Systems • Routing Protocol <ul style="list-style-type: none"> - RIP - OSPF | <ul style="list-style-type: none"> • Mapping OSI to TCP/IP model • Discussing the IP address format • Comparing public and private IP addresses • Discussing the role of DNS • Discussing the functions of RIP and OSPF | <ul style="list-style-type: none"> • Internet • Network simulators • Google drive • Drop box • Microsoft 365 • Printed modules |

| | | | | |
|--|---|---|--|--|
| | <ul style="list-style-type: none">• differentiate cloud service models• describe cloud types | <ul style="list-style-type: none">• Cloud Service Models<ul style="list-style-type: none">- Software as a Service (SaaS)- Platform as a Service (PaaS)• Cloud Types<ul style="list-style-type: none">- Public- Private | <ul style="list-style-type: none">• Discussing cloud service models• Using cloud services such as Google and Microsoft services• Comparing cloud types | |
|--|---|---|--|--|

LEVEL 3 Continued...

| SKILL/TOPIC | OBJECTIVE Learners should be able to: | CONTENT (KNOWLEDGE, SKILLS, ATTITUDES) | SUGGESTED LEARNING ACTIVITIES AND NOTES | SUGGESTED LEARNING RESOURCES |
|----------------------------------|---|--|--|--|
| 8.3 Computer Architecture | <ul style="list-style-type: none"> • explain the principle of operation of passive and active electronic components • describe the Von Neumann and Harvard Architecture • explain the use of buses • demonstrate the use of logic gates • describe the functions of processor components • identify factors affecting processor speeds • explain the importance of pipelining • describe the fetch-decode-execute cycle • identify types of interrupts • justify why computers use interrupts • explain addressing modes • construct an architectural design for a given scenario | <ul style="list-style-type: none"> • Hardware Architecture <ul style="list-style-type: none"> - Basic Electronics (Passive and Active Components) - Von Neumann Architecture - Harvard Architecture - Buses <ul style="list-style-type: none"> ○ control ○ address ○ data - Logic Gates <ul style="list-style-type: none"> ○ NAND ○ NOR ○ XOR ○ XNOR - Processor Components <ul style="list-style-type: none"> ○ CU ○ ALU ○ Registers (Program Counter, Memory Data Register, Memory Address Register, Index Register, Current Instruction Register, Status register accumulator) - Pipelining - Fetch-Decode-Execute cycle - Interrupts <ul style="list-style-type: none"> ○ Internal ○ External ○ Software - Addressing Modes <ul style="list-style-type: none"> ○ Direct, Indirect, Immediate, | <ul style="list-style-type: none"> • Testing and replacing electronic components • Drawing the Von Neumann and Harvard Architectural models • Distinguishing between the Von Neumann and Harvard Architecture • Discussing the use of buses • Drawing and integrate buses • Designing logic circuits • Constructing truth tables of up to three inputs • Calculating the outcome from a logic circuit given the inputs • Discussing the functions of processor components • Drawing processor components • Discussing factors affecting processor speeds • Demonstrating ways of improving processor performance • Discussing the concept of pipelining • Illustrating the fetch-decode-execute cycle • Distinguishing types of computer interrupts • Discussing computer interrupts | <ul style="list-style-type: none"> • Multimedia • Print media • Internet • Electronic components • Test instruments • Breadboard /Circuit board • Design tools such as Ms Visio, Smartdraw, E-draw • |

| | | | | |
|--|--|--|---|--|
| | | <p>Indexed and Relative</p> <ul style="list-style-type: none"> • Software Architecture - Architectural design | <ul style="list-style-type: none"> • Distinguishing addressing modes • Drawing an architectural design from user requirements | |
|--|--|--|---|--|

LEVEL 3 Continued...

| SKILL/TOPIC | OBJECTIVE Learners should be able to: | CONTENT (KNOWLEDGE, SKILLS, ATTITUDES) | SUGGESTED LEARNING ACTIVITIES AND NOTES | SUGGESTED LEARNING RESOURCES |
|--------------------------------|--|--|--|---|
| 8.4 Security and Ethics | <ul style="list-style-type: none"> • explain the difference between data privacy and integrity • analyze common threats and vulnerabilities on computer systems • identify sources of vulnerability • describe how data is kept safe during storage and transmission • evaluate tools used to eliminate vulnerabilities • identify relevant ICT legislative and regulatory frameworks • identify risks to Computer Systems • explore techniques and practices of risk management • outline the importance of securing data at off-site locations (cloud computing) • identify code of ethics and professional practices in the | <ul style="list-style-type: none"> • Data privacy and Integrity • Common threats and software vulnerabilities to computer systems • Protection, access control, and authentication of data • Legislation on computer security and crimes • Disaster Recovery methods • Risk Management techniques <ul style="list-style-type: none"> • Code of ethics at the workplace • Business ethics such as: | <ul style="list-style-type: none"> • Conducting case studies on different attack scenarios • Evaluating major counter measures to software and system attacks • Listing tools used to eliminate vulnerabilities in computer systems • Carrying out system security administration tasks such as creating accounts, setting permissions • Describing relevant ICT legislative and regulatory frameworks • Conducting case studies on Risk analysis and management • Formulating a disaster recovery plan in case of system failure: • Applying code of ethics and professional practices in the Computing field • Discussing the dangers and | <ul style="list-style-type: none"> • ICT tools • Antivirus software • Firewalls • Anti-spyware • Data Protection Act • Print and Electronic Media/modules / guides • Encryption tools • Second Science Technology and Innovation Policy of Zimbabwe (2012) • Copyright Act (Chapter 26:1) • Anti-plagiarism software such as Turn-it- in, Viper • Expert Guest such as EMA personnel |

| | | | | |
|--|---|---|---|--|
| | <p>Computing field</p> <ul style="list-style-type: none"> • demonstrate ethical practices in Computer Science • discuss security policies, laws and computer crime • identify environmental impact from computer use and disposal • analyse the ethical impact of existing and emerging technologies • explain the attributes of business ethics | <ul style="list-style-type: none"> - Fairness - Firmness - Honesty - Self- motivation - Integrity (unhu/ubuntu/Vumunhu) • Impact of social media • Security Policies • Laws and Computer Crime: <ul style="list-style-type: none"> - Types - Control measures • Environmental laws and issues <ul style="list-style-type: none"> - E-waste management <ul style="list-style-type: none"> o 3Rs (Recycle, Reuse, Reduce) • Impact of social media • Ethical principles | <p>ills from misuse of social media</p> <ul style="list-style-type: none"> • Discussing the attributes of business ethics in computer science • Discussing relevant ICT legislative and regulatory frameworks • Discussing the methods of recycling e-waste • Investigating the legal, social, and ethical implications of both existing and emerging technologies • Applying code of ethics and professional practices in Computing field | |
|--|---|---|---|--|

Level 3 Continued...

| SKILL/TOPIC | OBJECTIVE Learners should be able to: | CONTENT (KNOWLEDGE, ATTITUDES) | SKILLS, SUGGESTED LEARNING ACTIVITIES AND NOTES | SUGGESTED LEARNING RESOURCES |
|---------------------------------|---|---|--|--|
| 8.5 Software Development | <ul style="list-style-type: none"> • identify the opportunity for software development • outline fact finding techniques • apply SDLC stages in software development • produce documentation for each stage • Explain the concept of software process models | <ul style="list-style-type: none"> • SDLC (Waterfall Model) <ul style="list-style-type: none"> - Preliminary investigation - Feasibility study - System Analysis - System design - System development - Testing - Implementation - Maintenance and Evaluation | <ul style="list-style-type: none"> • Selecting project areas in groups • Discussing fact finding techniques • Applying SDLC stages in a group project • Creating documentation for each stage • Presenting their group projects | <ul style="list-style-type: none"> • ICT tools • National ICT Policy |

| | | | | |
|---|--|--|--|--|
| | <ul style="list-style-type: none"> • Develop a project using process models | <ul style="list-style-type: none"> • Object Oriented Methodology (OOM) • Prototyping | <ul style="list-style-type: none"> • Discussing software process models • Applying software process models to a project • Applying software process models to a project | |
| 8.6 Algorithm Design and Data Structures | <ul style="list-style-type: none"> • analyse algorithms for a given situation • design algorithms for a given situation • demonstrate familiarity with standard algorithms • distinguish between dynamic and static data structures • perform operations on binary trees and arrays • outline primitive data types • use example to demonstrate recursion | <ul style="list-style-type: none"> • Pseudocode structures <ul style="list-style-type: none"> - Selection - Repetition /Iteration • Standard algorithms • Sorting algorithms <ul style="list-style-type: none"> - Bubble sort - Quick sort • Searching algorithms <ul style="list-style-type: none"> - Linear - Binary search • Data structures <ul style="list-style-type: none"> - Dynamic (binary tree) - Static (array) • Binary trees and array operations <ul style="list-style-type: none"> - Sorting - Deletion - Insertion - Searching • Primitive data types <ul style="list-style-type: none"> - Integer - Floating point - Character - Boolean • Recursion | <ul style="list-style-type: none"> • Formulating trace tables for given algorithms • Developing algorithms • Researching on standard algorithms • Using standard algorithms to solve problems • Performing binary operation on arrays and binary trees • Discussing primitive data types • Writing algorithms for recursive functions | <ul style="list-style-type: none"> • ICT tools • National ICT Policy • Smart Draw • Internet • Multimedia Tutorials |

Level 3 Continued...

| SKILL/TOPIC | OBJECTIVE Learners should be able to: | CONTENT (KNOWLEDGE, SKILLS, ATTITUDES) | SUGGESTED LEARNING ACTIVITIES AND NOTES | SUGGESTED LEARNING RESOURCES |
|------------------------|---|---|--|---|
| 8.7 Programming | <ul style="list-style-type: none"> • describe the features of high level languages • use the features of programming languages • use programming constructs in structured programming • manipulate an array • read/write to a file • design input and output interface • outline features of OOP • design games and mobile applications | <ul style="list-style-type: none"> • Types of high level languages <ul style="list-style-type: none"> - Imperative/procedural - Declarative - General purpose - Special purpose - Object Oriented Programming (OOP) • Programming languages features <ul style="list-style-type: none"> - Constants - Variables - Expressions - Statements - Control structure - Block structure - Variables (local and global) • Structured programming <ul style="list-style-type: none"> - Nested loops (up to 3) - Iteration - Multiple Selection • Functions and procedures <ul style="list-style-type: none"> - Parameter passing <ul style="list-style-type: none"> o by value o by reference • Arrays <ul style="list-style-type: none"> - 1-Dimensional - 2-Dimensional • File handling • Interface design <ul style="list-style-type: none"> - User profiling - Input and output design | <ul style="list-style-type: none"> • Describing characteristics and features of high level languages • Using the features of programming languages • Applying programming constructs in structured programming • Developing programs using functions • Using arrays in a program • Reading from and writing to a file • Constructing input and output designs • Implementing classes • Demonstrating the use of a singleton • Designing games with indigenous orientation and mobile device applications | <ul style="list-style-type: none"> • Programming tools such as VB.Net, Java, Python, PHP • Multimedia Tutorials • Expert Guest • Animation software such as Adobe Creative Software |

| SKILL/TOPIC | OBJECTIVE Learners should be able to: | CONTENT (KNOWLEDGE, ATTITUDES) SKILLS, | SUGGESTED LEARNING ACTIVITIES AND NOTES | SUGGESTED LEARNING RESOURCES |
|----------------------|---|--|---|--|
| | <ul style="list-style-type: none"> • | <ul style="list-style-type: none"> • Object Oriented Programming <ul style="list-style-type: none"> - Classes - Encapsulation - Polymorphism - Inheritance • Advanced Programming <ul style="list-style-type: none"> - Mobile applications • Gaming | <ul style="list-style-type: none"> • | <ul style="list-style-type: none"> • |
| 8.8 Databases | <ul style="list-style-type: none"> • describe the file based approach database systems • outline features of a DBMS • describe features of relational database which address limitations of a file based approach • write SQL commands • develop interfaces and queries using DBMS tools • access data in a database through a high level language • design database applications using Entity relationship diagrams (ERDs) • convert ERDs to relational databases schema in standard normal form • normalise database tables up to second normal form | <ul style="list-style-type: none"> • File based database systems • Database Management Systems (DBMS) <ul style="list-style-type: none"> - Data Definition Language(DDL) - Structured Query Language(SQL) - Data Manipulation Language (DML) • Relational database modeling <ul style="list-style-type: none"> - Analysis - Normalisation • Database management | <ul style="list-style-type: none"> • Discussing the characteristics of a file based approach database systems • Describing the features of a DBMS • Discussing the features of relational databases • Using SQL commands to create a database table • Creating SQL code to query or modify data in database tables • Developing interfaces using DBMS tools • Linking high level languages with databases • Constructing ERDs for database applications • Converting ERDs to relational databases schema in standard normal form | <ul style="list-style-type: none"> • Database software such as MySQL • Design tools such as Ms Visio, Smartdraw, E-draw • DBMS software tools • Modules /print media/tutorials |

| | | | | |
|--|--|--|--|--|
| | | | <ul style="list-style-type: none">• Normalising database tables up to second normal form | |
|--|--|--|--|--|

Level 3 Continued...

| SKILL/TOPIC | OBJECTIVE Learners should be able to: | CONTENT (KNOWLEDGE, ATTITUDES) SKILLS, | SUGGESTED LEARNING ACTIVITIES AND NOTES | SUGGESTED LEARNING RESOURCES |
|-------------------------|---|---|---|---|
| 8.9 Enterprising | <ul style="list-style-type: none"> • identify areas where computer science is applied • evaluate the importance of e-Business • explain the elements of marketing • design an ICT related business proposal • describe the role of Intellectual Property Rights • analyse global trends in the field of computing | <ul style="list-style-type: none"> • Application areas of Computer Science such as: <ul style="list-style-type: none"> - Engineering - Research and Development - Agricultural Mechanisation - E- learning - Telecommunications • Business Proposal Development • E-Business <ul style="list-style-type: none"> - Financial literacy - Marketing and business strategies, skills and techniques • Intellectual Property Right <ul style="list-style-type: none"> - Patents and Trademarks - Plagiarism and Copyrights issues • Global trend analysis | <ul style="list-style-type: none"> • Appreciating the role of computer science in various sectors • Performing basic accounting procedures for the business operation • Designing a viable ICT business plan • Discussing the 4P's of marketing • Discussing how to safeguard innovations • Conducting research on global trends in the field of computing • Conducting educational tours to local industries and business centres | <ul style="list-style-type: none"> • ICT tools • Expert Guest • Second Science Technology and Innovation Policy of Zimbabwe (2012) • Marketing plan templates • Business plan templates • Statistical tools |

9.0 ASSESSMENT

In order to have a holistic assessment, learners will be assessed in the aspects of continuous and summative assessment with each contributing to the learner's final grade.

9.1 Assessment Objectives

Learners will be assessed in the following areas:

(i) Knowledge and Understanding

Learners should be able to:

- describe the use of computer science in a range of information processing systems
- outline the functions of the hardware and software components in a computer system
- explain the use of SDLC in software development
- explain the social, economic, ethical and legal effects of computer use on individuals, organization and society

(ii) Problem Solving

Learners should be able to:

- apply programming concepts to solve real life problems
- design computer systems based on user requirements specifications
- design an appropriate algorithm for a given scenario

(iii) Communication Skills

Learners should be able to:

- develop an understanding of the component parts of computer systems and how they inter-relate including software, data, hardware, communications and people
- interpret and present information in a variety of forms
- clarify source code through comments

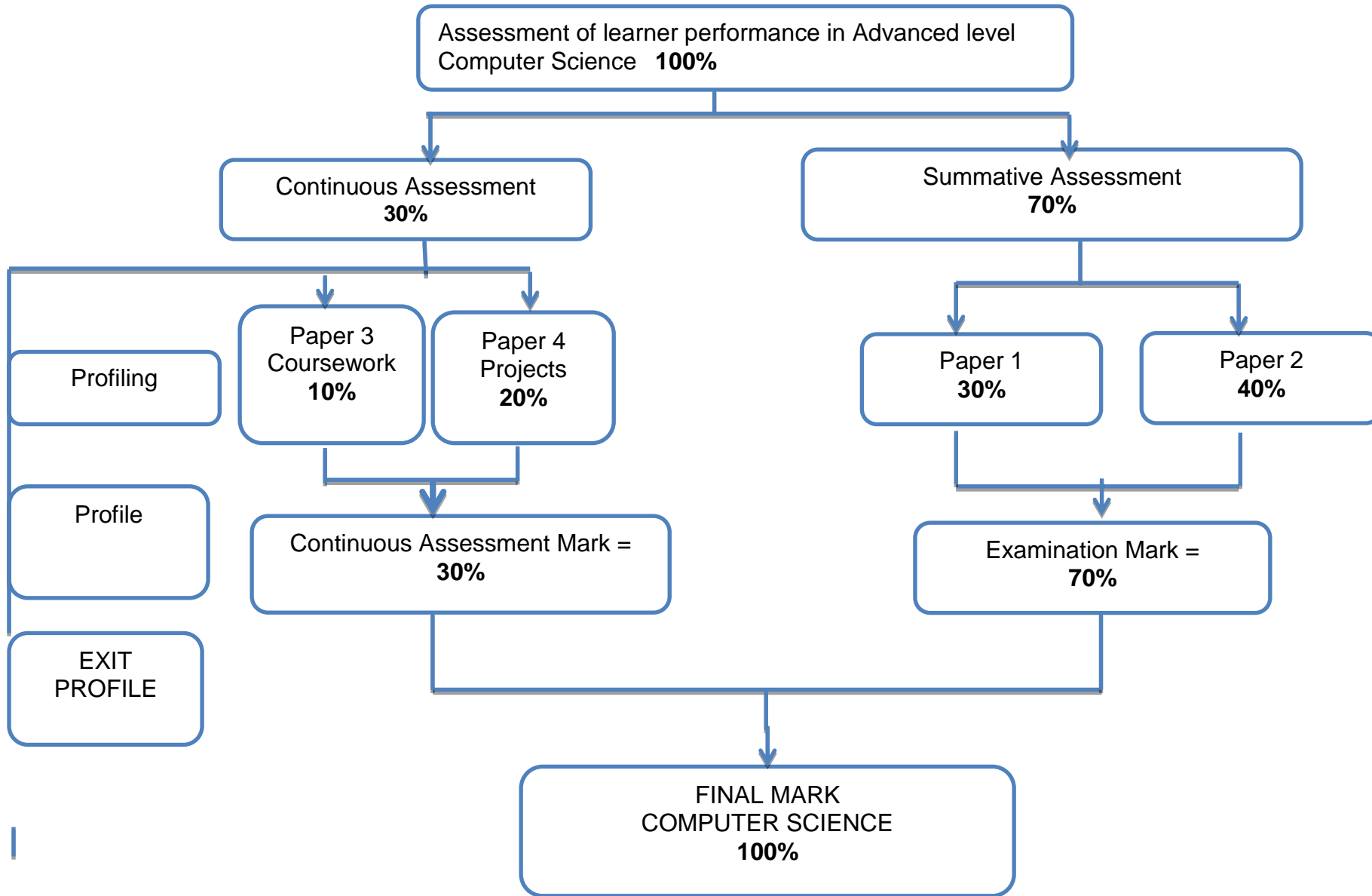
(iv) Practical Skills

Learners should be able to:

- design, develop and document computer systems to solve problems
- demonstrate proficiency in the creation, design and implementation of computer solutions using prescribed programming packages
- design logic circuit for a given task

9.2 Scheme of Assessment

Computer Science learning area will be examined through continuous and summative assessment as follows.



The Scheme of Assessment is intended to encourage positive achievement by all learners. The subject will be examined in 4 papers as shown in the table below.

| Paper | Form of Assessment | Type of Paper | Duration | Total marks | Weighting (%) |
|--------------|-----------------------|---------------|----------|-------------|---------------|
| 1 | Summative Assessment | Free Response | 3 Hours | 100 | 30 |
| 2 | | Practical | 3 Hours | 100 | 40 |
| 3 | Continuous assessment | Coursework | 5 terms | 100 | 10 |
| 4 | | Project | 4 terms | 100 | 20 |
| TOTAL | | | | 400 | 100 |

9.3 Specification Grid

(a) Content distribution

PAPER 1

| TOPIC | WEIGHTING (30%) |
|--------------------------------------|-----------------|
| Data Representation | 15 |
| Networking | 20 |
| Computer Architecture | 20 |
| Security and Ethics | 10 |
| Algorithm Design and Data Structures | 15 |
| Databases | 10 |
| Enterprising | 10 |
| TOTAL | 100 |

PAPER 2

| SECTION | WEIGHTING (40%) |
|---|------------------------|
| Computer Architecture | 30 |
| Programming, Algorithm design and Databases | 40 |
| Networking | 30 |
| TOTAL | 100 |

(b) Skills distribution

All internal and external tests on cognitive skills (theory) shall be 40% knowledge and understanding and 60% problem solving. All internal and external practical tests shall be 100% practical skills. Thus the weighting of questions based on the skills for the subject, will be as follows:

| SKILL | Paper 1(%) | PAPER 2(%) |
|-----------------------------|-------------------|-------------------|
| Knowledge and Understanding | 30 | 10 |
| Problem Solving | 50 | 40 |
| Practical Skills | 20 | 50 |
| TOTAL | 100 | 100 |

9.4 Paper Descriptions

Paper 1: Theory (100 Marks)

The paper consists of 10 to 12 compulsory questions.

Paper 2: Practical (100 Marks)

The paper consists of 3 sections with 6 practical questions where candidates answer 1 question from each section.

Paper 3: Continuous Assessment (Coursework) {100 marks}

Coursework is made up of 7 items which comprises of five practical assignments and two theory tests. Practical assignments and tests are set, marked and recorded internally by teachers. Each of these assignments and tests should match the skills distribution table given above. The internal practical assignments are spaced equitably from the beginning of the level to the end. The internal theory coursework tests should be written within the. All marked practical and theory scripts including the compiled mark schedule will be submitted to ZIMSEC.

Paper 4: Project (100 marks)

Examination Centres are advised to encourage their candidates to identify real life problems within their environment and develop solutions in line with the project guide lines. This will require candidates to have practical computer science skills with a bias towards designing, electronic, programming or networking based projects. Knowledge of programming language syntax will be examined in this project report. The higher ability candidates are to be encouraged to extend their practical programming beyond the scope of these tasks. The project work should be carried out from term two .The marked project reports including the compiled mark schedule will be submitted to ZIMSEC.

10.0 GLOSSARY/APPENDICES

APPENDIX I: GLOSSARY OF TERMS USED IN QUESTION PAPERS

It is hoped that the glossary will be helpful to learners as a guide. The glossary has been deliberately kept brief not only with respect to the number of terms included but also to the descriptions of their meanings. Learners should appreciate that the meaning of a term must depend in part on its context.

- | | | |
|----|----------------------------|--|
| 1 | Define | is intended literally, only a formal statement or equivalent paraphrases being required. |
| 2 | State | implies a concise answer with little or no supporting argument e.g. numerical answer that can readily be obtained by inspection |
| 3 | List | requires a number of points generally each of one word with no elaboration, where a number of points is specified this should not be exceeded. |
| 4 | Explain | may imply reasoning or some reference to theory depending on the context |
| 5 | Describe concept | requires the candidate to state in words (using diagrams where appropriate) the main points of the |
| 6 | Outline | implies brevity that is restricting the answer to given essentials |
| 7 | Predict/deduce | the candidate is expected to produce the expected answer by making a logical connection between other pieces of information |
| 8 | Suggest | it is used in two main contexts that is either to imply that there is no unique answer or to imply that learners are expected to apply their general knowledge |
| 9 | Find | is a general term that may variously be interpreted as calculate, measure, determine etc |
| 10 | Determine | often implies that the quantity concerned cannot be measured directly but is obtained by calculation |

APPENDIX II: ACRONYMS

| | |
|----------------|---|
| ASCII- | American Standard Code for Information Interchange. |
| BCD | Binary-Coded Decimal. |
| DBMS | Database Management Systems |
| DNS | Domain Name System |
| EBCDIC | Extended Binary Coded Decimal Interchange Code |
| ICT | Information and Communications Technology - or Technologies |
| IP | Internet Protocol |
| OOM | Object Oriented Methodology |
| OOP | Object Oriented Programming |
| OSI | Open Systems Interconnection |
| OSPF | Open Shortest Path First |
| PaaS | Platform as a Service |
| RIP | Routing Information Protocol |
| SaaS | Software as a Service |
| SDLC | System Development Life Cycle |
| UNICODE | Universal trunk - out-of-service CODE |
| EMA | Environmental Management Agent |

APPENDIX III: PRACTICAL ASSESSMENT GUIDELINES

Computer Science is a practical subject and a range of practical exercises should complement the study of the practical parts of the syllabus. It is recommended that learners should be exposed practical lessons. It is also recommended that the computer-learner ratio be 1:1. Where possible, machines should be loaded with integrated packages to facilitate easy importing and exporting of documents.

Practical Examination

The practical examination session should be invigilated by the any teacher from any department and Computer Science teacher(s) should be available for technical support.

Project Guide (ZIMSEC to consider separate guidelines for candidates doing hardware related projects)

The project must not exceed 50 pages excluding appendices. The project must include the following layout

1. Cover Page.
2. Table of Contents.
3. Appendices.

The Appendices include any two of the following.

- Sample of completed questionnaires
- Sample of interview questions with respondent answers
- Sample documents

NB the project must be spiral bound

SECTION A (25 marks)

Selection, Investigation and Analysis

- Define a problem
 - Choice of problem area and background analysis. [3]
- Investigation of the current system
 - Data analysis using DFDs, flow charts and ERDs
 - Research instruments e.g. questionnaire, record inspection, interviews and observation.
 - Identify problems with the current system. [5]
- Feasibility study [5]

- Requirements specification
 - User
 - Software
 - Hardware.[4]
- Aims and objectives. [5]
- Evidence that the research has been carried out. [3]
 - Examples are filled in questionnaires, interviews with respondent answers, sample documents and write up on observation.

SECTION B (25 marks)

Design

- Consideration of alternative method. [3]
 - Justification of proposed solution [2]
- Input design
 - Appropriate data capture forms and screen layouts [4]
- Data Structures/File design [5]

OR

- Object Oriented Design
 - Class diagrams
 - Use Case diagrams
 - Sequence diagrams
- Overall Plan [3]
- Output Design
 - Specification and design of the required output
 - Interface design (on screen commands) [5]
- Test Strategy/Test Plan
 - Select test plan and justify [3]

SECTION C (25 marks)

Software Development

- Techniques that improve the structure, appearance and clarity of the code that is:
 - Procedures
 - Functions
 - Scope of variables (local and global)
 - Use of comments
 - Blank lines
 - Indentation

[10]

- Technical documentation
 - Algorithms
 - pseudo codes
 - flowcharts for modules
 - codes/program listings

[7]

- User documentation
 - Installation
 - Starting the systems
 - Navigation of the system
 - Exiting the system

[8]

SECTION D (15 marks)

Testing and Evaluation

- User Testing
 - Design and select test data
 - Test for standard, extreme and abnormal/invalid data
 - Evidence of testing to be shown through sample runs and error messages [5]

- System Testing
 - Ease of use
 - clarity of instruction to the user
 - Reliability
 - produce reliable results, there should be no bugs
 - Effectiveness
 - The system should work efficiently
 - Produce results with minimum delay [5]

- Evaluation Limitations of the system
 - Extent of success in meeting the system objectives as stated in the system requirement specification
 - Achievements
 - Limitations
 - Evaluate results against the system objectives – achievements and limitations
 - Opportunities for future development [5]

SECTION E (10 marks)

General Expectations

- Depth of Knowledge and Understanding
 - Reflects the degree of computing in the project
 - Is the code fairly standard?
 - Different techniques implemented [2]

- Degree of Originality
 - Imagination and innovation
 - Has an attempt been made to do something different/unique? [2]

- Overall conduct of the project
 - Is the work carefully organized? The degree of help to be reflected [1]

- Quality of the completed report
 - Written report should be easy to follow
 - Defined sections, page numbers and an index. [5]

APPENDIX IV: RESOURCES AND EQUIPMENT

Infrastructure and Equipment

For a school to run the Computer Science Syllabus for examination purposes, the under listed infrastructure and equipment need to be in place

Computer Laboratory

Personal Computers to accommodate 1 learner per computer and a printer

Computer Desks and Chairs to accommodate the number of learners

Dustless Displays for the Teacher (securely-mounted Whiteboard, LCD projector)

Theory Classroom

Classroom furniture to accommodate the learners

Writing Surface for the Teacher (e.g. securely-mounted Whiteboard, LCD projector)

In both the above cases, there should be adequate lighting and ventilation. Element of inclusivity should be taken into account.

APPENDIX V: SUGGESTED REFERENCE BOOKS

It should be noted that specifying a limited list of textbooks is difficult as new titles are being availed all the time. Teachers are therefore encouraged to consult other books in order to adequately cover the whole syllabus. However, below is a suggested book list which serves the purpose of being a reference guide.

- Langfield. S & Heathcote, '**A' Level Computing, 5th Edition**, Payne-Gallway Publishers, 2004.
- Ray Bradley, **Understanding Computer Science**.
- Sylvia Langfield & Dave Duddell, **Cambridge International AS and A Level Computer Science Coursebook**, 2015.
- British Computer Society (2005), **The BCS Glossary ICT and Computer Terms**, McMillan, UK
- Kevin Bond, **A 'Level Computer Science**.