ZIMBABWE

MINISTRY OF PRIMARY AND SECONDARY EDUCATION

COMPUTER SCIENCE SYLLABUS

FORMS 5 - 6

2024 - 2030

Curriculum Development and Technical Services
P. O. Box MP 133
Mount Pleasant
Harare

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
- Ministry of Higher and Tertiary Education, Innovation, Science and Technology Development (MoHTEISTD)
- Zimbabwe School Examinations Council (ZIMSEC)
- United Nations Children's Education Fund (UNICEF)
- United Nations Educational, Scientific and Cultural Organization (UNESCO)

CONTENTS

1.0 PREAMBLE

1.1 Introduction

The Heritage-Based Computer Science syllabus follows the Science, Technology, Engineering and Mathematics (STEM) do- main and is designed to cover forms 5 and 6 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 Heritage-Based Computer Science syllabus covers theory and practical activities for learners in forms 5 and 6. 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 eco- nomically viable solutions

1.5 Cross-Cutting themes

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

- Climate Change
- Enterprise skills
- Health and wellbeing
- Environmental issues
- Disaster Risk management
- ICT
- Child rights and responsibilities

2.0 PRESENTATION OF SYLLABUS

The Heritage-Based Computer Science syllabus is presented as one document for Forms 5 and 6.

3.0 AIMS

The Heritage-Based Computer Science 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 Heritage-Based 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 subject should be allocated at least 12 periods of 40 minutes per week, 4 theory periods and 8 practical periods. 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 System Development Life Cycle (SDLC)
5.6 Algorithm Design and Data Structures
5.7 Programming
5.8 Databases
5.9 Enterprising

7.0 SCOPE AND SEQUENCE CHART

TOPIC	FORM 5	FORM 6
7.1 Data Representation	• Number bases • Floating point arithmetic • Computer arithmetic errors • Data representation	
7.2 Computer Architecture	• Hardware Architecture	• Software Architecture
7.3 Networking	• 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
7.4 Systems Development Life Cycle (SDLC)	• Systems Development Life Cycle (SDLC- Waterfall Model)	• Object Oriented Methodology • Prototyping

TOPIC	FORM 5	FORM 6
7.5 Security and Ethics	• 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 • Impact of social media • Ethical principles
7.6 Algorithm Design and Data Structures	• Pseudocode structures • Standard algorithms • Data structures	• Binary trees and array operations • Primitive data types • Recursion
7.7 Programming	• Programming language features • Structured programming • Functions and procedures • Arrays • File handling • Interface design	• Object Oriented Programming • Advanced Programming
7.8 Databases	• File based database systems • Database Management Systems (DBMS) • Relational Database modeling • Database management	
7. 9 Web Application Development	• Client side web development	• Server side web development
7.10 Enterprising	• E-Business • Intellectual Property Rights • Application areas of Computer Science • Business Proposal Development	• E-Commerce • Telecommunications • Business and enterprise

8.0 COMPETENCY MATRIX

FORM 5

SKILL/TOPIC	OBJECTIVE Learners should be able to:	CONTENT (knowledge, skills, values and attitudes)	SUGGESTED LEARNING ACTIVITIES AND NOTES	SUGGESTED LEARNING RESOURCES
8.1 Data Representation	• convert number bases • multiply and divide binary numbers • normalize floating point binary numbers • represent data and character sets • distinguish arithmetic errors • interpret arithmetic errors	• Number bases - binary - octal - denary - hexadecimal - 2s Compliment • Floating point arithmetic - normalization of floating-point numbers • Computer arithmetic errors - overflow - underflow • Data representation - Character sets - ASCII - UNICODE - BCD - EBCDIC	• 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	• Calculators • Print and electronic media • Data set tables • ICT tools • Internet
8.2 Networking	• 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	• OSI Model • TCP/IP Model • IP Addressing • Domain Name Systems • Routing Protocol - RIP - OSPF	• 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	• Internet • Network simulators

SKILL/TOPIC	OBJECTIVE Learners should be able to:	CONTENT (knowledge, skills, values and attitudes)	SUGGESTED LEARNING ACTIVITIES AND NOTES	SUGGESTED LEARNING RESOURCES
8.3 Computer Architecture	• 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	• Hardware Architecture - Basic Electronics (Passive and Active Components) - Von Neumann Architecture - Harvard Architecture - Buses ○ control ○ address ○ data - Logic Gates ○ NAND ○ NOR ○ XOR ○ XNOR - Processor Components ○ 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 ○ Internal ○ External ○ Software - Addressing Modes ○ Direct, Indirect, Immediate,	• 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	• Multimedia • Print media • Internet • Electronic components • Test instruments • Breadboard /Circuit board

	interrupts • explain addressing modes	Indexed and Relative	• Discussing computer interrupts • Distinguishing addressing modes	

8.4 Security and Ethics	• 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 Computing field	• 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 • Code of ethics at the workplace • Business ethics such as: - Fairness - Firmness - Honesty - Self- motivation	• 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	• ICT tools • Antivirus software • Firewalls • Anti-spyware • Data Protection Act • Print and Electronic Media • Encryption tools • Second Science Technology and Innovation Policy of Zimbabwe (2012)

	• demonstrate ethical practices in Computer Science	- Integrity (unhu/ubuntu/Vumunhu) • Impact of social media	• Discussing the dangers and ills from misuse of social media • Discussing the attributes of business ethics in computer science	
8.5 Software Development	• identify the opportunity for software development • outline fact finding techniques • apply SDLC stages in software development • produce documentation for each stage	• SDLC (Waterfall Model) - Preliminary investigation - Feasibility study - System Analysis - System design - System development - Testing - Implementation - Maintenance and Evaluation	• 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	• ICT tools • National ICT Policy
8.6 Algorithm Design and Data Structures	• analyse algorithms for a given situation • design algorithms for a given situation • demonstrate familiarity with standard algorithms • distinguish between dynamic and static data structures	• Pseudocode structures - Selection - Repetition /Iteration • Standard algorithms • Sorting algorithms - Bubble sort - Quick sort • Searching algorithms - Linear - Binary search • Data structures - Dynamic (binary tree) - Static (array)	• Formulating trace tables for given algorithms • Developing algorithms • Researching on standard algorithms • Using standard algorithms to solve problems	• ICT tools • National ICT Policy

SKILL/TOPIC	OBJECTIVE Learners should be able to:	CONTENT (knowledge, skills, values and attitudes)	SUGGESTED LEARNING ACTIVITIES AND NOTES	SUGGESTED LEARNING RESOURCES
8.7 Programming	• describe the features of high-level languages	• Types of high level languages - Imperative/procedural - Declarative - General purpose - Special purpose - Object Oriented Programming (OOP)	• Describing characteristics and features of high-level languages	• Programming tools such as VB.Net, Java, Python • Multimedia Tutorials • Expert Guest
	• use the features of programming languages	• Programming languages features - Constants - Variables - Expressions - Statements - Control structure - Block structure - Variables (local and global)	• Using the features of programming languages	
	• use programming constructs in structured programming	• Structured programming - Nested loops (up to 3) - Iteration - Multiple Selection • Functions and procedures - Parameter passing o by value o by reference	• Applying programming constructs in structured programming • Developing programs using functions	
	• manipulate an array	• Arrays - 1-Dimensional - 2-Dimensional	• Using arrays in a program	
	• read/write to a file	• File handling	• Reading from and writing to a file	
	• design input and output interface	• Interface design - User profiling - Input and output design	• Constructing input and output designs	
8.8 Databases	• describe the file-based approach database systems • outline features of a DBMS	• File based database systems • Database Management Systems (DBMS) - Data Definition	• Discussing the characteristics of a file-based approach database systems	• Database software such as MySQL • Design tools such as MS Visio,

	• 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	Language (DDL) - Structured Query Language (SQL) - Data Manipulation Language (DML) • Relational database modeling - Analysis - Normalisation • Database management	• 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 • Normalising database tables up to second normal form	Smart draw, E-draw • DBMS software tools

8.9 Web Application Development	• Design parts of a website or web application that users interact with. • Create Forms • Create navigation tools to allow users to move between different pages. • Validate websites	• User Interface creation. - Headings - Paragraphs - Strike through - Graphics - Tables - Bullets and numbering - Text formatting • Form Controls - Label - Textbox - Radio button - Check box - Combo box - List box - Button • Drop down menus • Active links Debug webpages and websites	• Constructing web pages • Linking web pages to construct a website • Debugging websites	• HTML • Java Script • PHP • CSS • Text editors such as Notepad++, Visual Studio code, Vim, Atom • Internet AI applications

SKILL/TOPIC	OBJECTIVE Learners should be able to:	CONTENT (knowledge, skills, values and attitudes)	SUGGESTED LEARNING ACTIVITIES AND NOTES	SUGGESTED LEARNING RESOURCES
8.10 Enterprising	• identify areas where computer science is applied	• Application areas of Computer Science such as: - Engineering - Research and Development - Agricultural Mechanisation - E- learning - Telecommunications	• Appreciating the role of computer science in various sectors	• ICT tools • Expert Guest • Second Science Technology and Innovation Policy of Zimbabwe (2012)
		• Business Proposal Development	• Performing basic accounting procedures for the business operation	• Marketing plan templates • Business plan templates
	• evaluate the importance of e-Business	• E-Business - Financial literacy		• Statistical tools
	• explain the elements of marketing	- Marketing and business strategies, skills and techniques		
	• design an ICT related business proposal		• Designing a viable ICT business plan	
	• describe the role of Intellectual Property Rights	• Intellectual Property Right - Patents and Trademarks - Plagiarism and Copyrights issues	• Discussing the 4P's marketing • Discussing how to safeguard innovations	
	• analyse global trends in the field of computing	• Global trend analysis	• Conducting research on global trends in the field of computing • Conducting educational tours to local industries and business centres	

FORM 6

SKILL/TOPIC	OBJECTIVE Learners should be able to:	CONTENT (knowledge, skills, values and attitudes)	SUGGESTED LEARNING ACTIVITIES AND NOTES	SUGGESTED LEARNING RESOURCES
8.11 Networking	• differentiate cloud service models • describe cloud types	• Cloud Service Models - Software as a Service (SaaS) - Platform as a Service (PaaS • Cloud Types - Public - Private	• Discussing cloud service models • Using cloud services such as Google and Microsoft services • Comparing cloud types	• Google drive • Drop box • Microsoft 365
8.12 Algorithm Design and Data Structures	• perform operations on binary trees and arrays • outline primitive data types • use example to demonstrate recursion	• Binary trees and array operations - Sorting - Deletion - Insertion - Searching • Primitive data types - Integer - Floating point - Character - Boolean • Recursion	• Performing binary operation on arrays and binary trees • Discussing primitive data types • Writing algorithms for recursive functions	• ICT tools • Smart Draw • Internet • Multimedia Tutorials
8.13 Programming	• outline features of OOP • design games and mobile applications	• Object Oriented Programming - Classes - Encapsulation - Polymorphism - Inheritance • Advanced Programming - Mobile applications - Gamming	• Implementing classes • Demonstrating the use of a singleton • Designing games with indigenous orientation and mobile device applications	• Programming Tools such as Java, Python, VB.Net, PHP • Animation software such as Adobe Creative Software

SKILL/TOPIC	OBJECTIVE Learners should be able to:	CONTENT (knowledge, skills, values and attitudes)	SUGGESTED LEARNING ACTIVITIES AND NOTES	SUGGESTED LEARNING RESOURCES
8.14 Web Application Development	• Create a database • Connect a web site to a database. • Perform basic CRUD operations. • Publish a website on the Web	• Setup database and tables. • Database connectivity using server side language. • CRUD operations - Create - Retrieve - Update - Delete - • Website publishing	• Constructing database structure/ schema. • Connecting a website to a database. • Testing database connectivity. • Writing scripts to perform CRUD operations through a web interface. • Simulating hosting a website using a free platform.	• Database Management System (DBMS) such as MySQL, PostgreSQL, SQL Server, SQLite • HTML • Web browser • Java Script • PHP • CSS • Text editors such as Notepad++, Visual Studio code, Vim, Atom. • GitHub, Netify
8.15 Computer Architecture	• construct an architectural design for a given scenario	• Software Architecture - Architectural design	• Drawing an architectural design from user requirements	• Internet • Design tools such as MS Visio, Smart draw, E- draw
8.16 Software development	• explain the concept of software process models • develop a project using process models	• Object Oriented Methodology (OOM) • Prototyping	• Discussing software process models • Applying software process models to a project	• ICT Tools

8.17 Security and Ethics	• 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	• Security Policies • Laws and Computer Crime: - Types - Control measures • Environmental laws and issues - E-waste management o 3Rs (Recycle, Reuse, Reduce) • Impact of social media • Ethical principles	• 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	• ICT tools • Copyright Act (Chapter 26:1) • Anti-plagiarism software such as Turn- it- in, Viper • Expert Guest such as EMA personnel

9.0 ASSESSMENT

The Heritage-based Computer Science syllabus learning area for form 5 and 6 shall be assessed through School Based Continuous Assessment (SBCA) and Summative Assessment (SA). These assessments shall be guided by the principles of inclusivity, practicability, authenticity, transparency, flexibility, validity and reliability. The principles are crucial for creating a supportive and effective learning environment that fosters growth and development in learners at secondary school level. Arrangements, accommodations and modifications shall be visible to enable candidates with special needs to access assessments.

This section covers the assessment objectives, the assessment model, the scheme of assessment, and the specification grid.

9.1 Assessment Objectives

By the end of the Heritage-based Computer Science Syllabus learning area for form 5 and 6 learners will be assessed on their ability to:

9.1.1 demonstrate an understanding of data representation in computer systems
9.1.2 apply the fundamental principles and concepts of computer science in algorithm design
9.1.3 demonstrate an understanding of computer architecture
9.1.4 apply ethical principles and standards in the field of computing
9.1.5 apply security measures to protect computer systems
9.1.6 show awareness of the data protection legislation and intellectual property rights
9.1.7 analyze technological changes and make informed decisions
9.1.8 apply programming concepts to solve real life problems
9.1.9 design computer systems based on user requirements and specifications

9.2 Assessment Model

Assessment of learners at Upper Secondary School Level for Heritage-based Computer Science Syllabus shall be both Continuous and Summative as illustrated in Figure 1. School Based Continuous Assessment shall include recorded activities from the School Based Projects done by the learners. The mark shall be included on learners' end of term and year reports. Summative assessment at school level shall include terminal examinations which are at the end of the term and year.

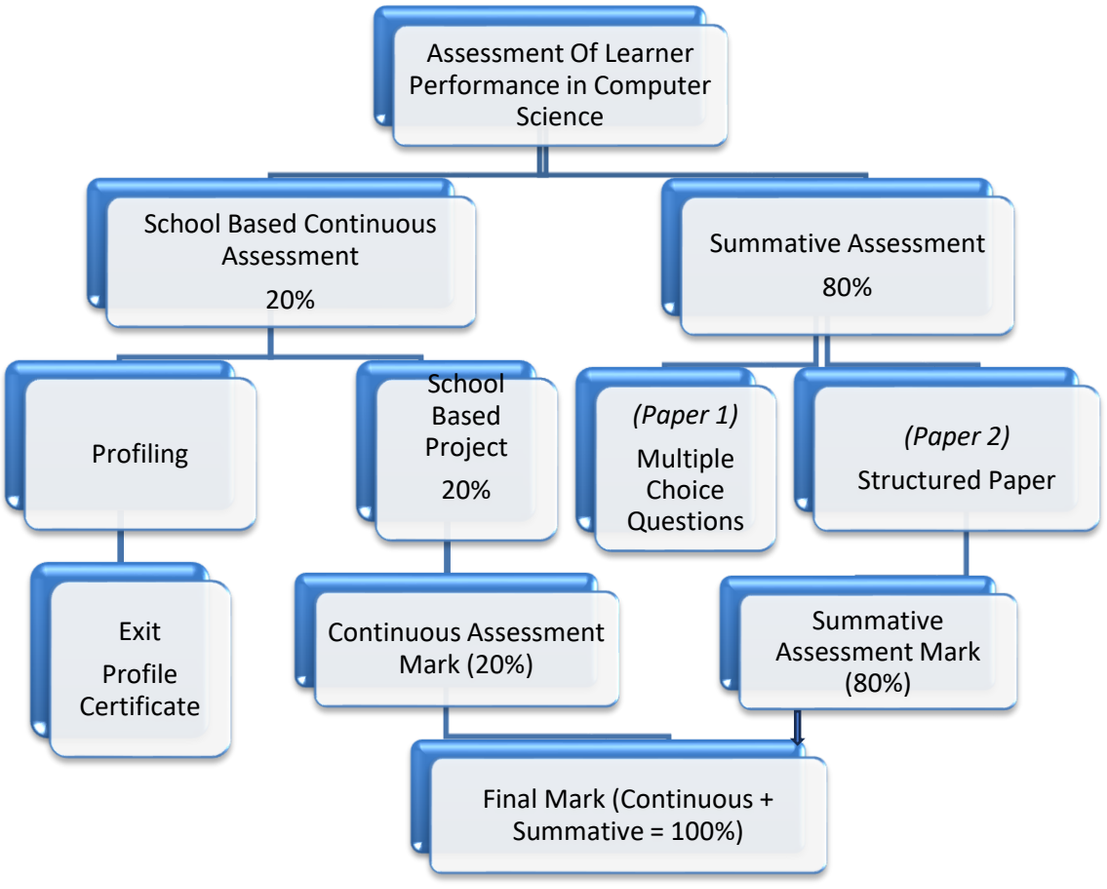

Fig. 1 Assessment Model

In addition, learners shall be profiled and learner profile records established. Learner profile certificates shall be issued for checkpoints assessment in schools as per the dictates of the Teacher's Guide to Learning and Assessment. The aspects to be profiled shall include learner's prior knowledge, values and skills, and subsequently the new competences acquired at any given point.

9.3 Scheme of Assessment

The Assessment Model shows that learners shall be assessed using both School Based Continuous Assessment and Summative Assessment for both School and ZIMSEC assessments.

The table shows the Scheme of Assessment where 20% is allocated to School Based Continuous Assessment and 80% to School or ZIMSEC Summative Assessment.

FORM OF ASSESSMENT	WEIGHTING
School Based Continuous Assessment	**20%**
Summative Assessment	**80%**
Total	**100%**

9.3.1 Description of School Based Continuous Assessment

Learners shall do one school-based project per form which contributes to 20% of the end of year final mark. The end of year summative assessment shall then contribute 80%. However, for ZIMSEC public examinations, two (2) school-based projects shall be considered as School Based Continuous Assessment at form 5 and 6. The two School Based Projects shall include those done during form 5 and 6 sessions. Each will contribute 10%.

9.3.1.1: School – Based Project Continuous Assessment Scheme

The Table given below shows the Learning and Assessment Scheme for the School Based Project.

Project Execution Stages	Project Stage Description	Timelines	Marks
1	Problem Identification	January	5
2	Investigation of related ideas to the problem/innovation	February	10
3	Generation of possible solutions	March	10
4	Selecting the most suitable solution	April-May	5
5	Refinement of selected solution	June	5
6	Presentation of the final solution	July	10
7	Evaluation of the solution and Recommendations	August-September	5
	TOTAL		**50**

9.3.2 Description of the ZIMSEC Summative Assessment

ZIMSEC Summative Assessment shall be a public examination at Form 6. The examination shall consist of two (2) papers of different weighting as shown in the table below. The Scheme of Assessment is intended to encourage positive achievement by all learners.

Paper	Type of Paper	Duration	Total marks	Weighting (%)
1	Free Response	3 Hours	100	30
2	Practical	3 Hours	100	50
TOTAL			**200**	**80**

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.

9.4 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 (50%)
Web Application Development	20
Programming and Algorithm design	50
Databases	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. The weighting of questions based on the skills for the learning area, will be as follows:

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

10.0 GLOSSARY/APPENDICES

APPENDIX I: GLOSSARY OF TERMS USED IN QUESTION PAPERS

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	requires the candidate to state in words (using diagrams where appropriate) the main points of the concept
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 to eight practical lessons per week. 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 message [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 needs 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.