

ZIMBABWE SCHOOL EXAMINATIONS COUNCIL



For Performance Measurement

ADVANCED LEVEL SYLLABUS

COMPUTING (9195)

(EXAMINATION PERIOD 2014 to 2017)

*** Available in the November Examinations only and not available to private candidates.

TABLE OF CONTENTS

Contents	Page
Preamble	2
Aims	2
Assessment Objectives	3
Assessment	4
Grade Descriptors	5
Methodology and Time Allocation	7
Methodology	7
Time Allocation	7
Section 1: Application of Computers and their Social and Economic Implications.....	9
Section 2: Systems Analysis and Design	11
Section 3: File Organisation and Database Concepts	12
Section 4: Data Structures, Types and Representations	14
Section 5: Algorithm Design and Programming Concepts	15
Section 6: Software & Hardware	19
Section 7: Architecture, Data Communications And Networks.....	21
Appendices	24
Appendix I: Acronyms	24
Appendix II: Project Guidelines/ Practical	26
Appendix III: Minimum Resources And Equipment Requirements	30
Appendix IV: Possible Career Opportunities	31
Appendix V: Reference Books And Journals	32

1.0 PREAMBLE

This syllabus is designed to provide a wide and deep study of the underlying principles, concepts, uses and limitations of computers as may be encompassed in an Advanced-level Computing subject. This is a two-year course that provides a useful foundation for pursuit of further education or branching to more specific and specialised fields of computing.

Learners will also be well equipped to appreciate and exploit future developments of modern technology and their effects in both the social and economic front.

It is envisaged that learners should gain an understanding of the fundamental principles of designing, using and implementing a wide range of information processing applications. Learners should not only understand the underlying mechanisms and methods which allow a computer to perform a required task but also how computers communicate. To reinforce an understanding of these central concepts of computing, the syllabus takes a more analytical and practical approach to computing. Learners will be required to apply skills learnt on concepts of a chosen programming language and carry out a project to a successful conclusion.

2.0 AIMS

The syllabus aims to enable learners to

- 2.1 develop an understanding of the nature and principles of computing in solving problems
- 2.2 acquire knowledge of the range of applications of computers including their social and economic impact
- 2.3 acquire practical skills of the main principles of systems analysis and design, problem formulation and planning of solutions using computers, and systematic methods of implementation, testing and documentation
- 2.4 develop an understanding of the ways in which data can be sensibly organised for processing by the computer
- 2.5 appreciate the organisation of computer systems including software, data, hardware and communication
- 2.6 show an awareness of the technological changes in the computer world
- 2.7 develop interest, enjoyment and confidence in the daily use of computers

3.0 ASSESSMENT OBJECTIVES

3.1 Knowledge And Understanding

Learners should be able to:

- 3.1.1 describe the use of computing in a range of information processing applications
- 3.1.2 explain the need for various forms of data organisation and processing to support the information requirements of particular applications
- 3.1.3 explain the systematic development of solutions to problems and the appropriate techniques for implementing such solutions
- 3.1.4 demonstrate an understanding of the characteristics of software, hardware and communication elements of the computer systems which allow effective solutions to be achieved

3.2 Problem Solving

Learners should be able to:

- 3.2.1 analyse an application and identify parts which are appropriate for a computer-based solution
- 3.2.2 comment on social, economic, ethical and legal effects of computer use on individuals, organisations and society
- 3.2.3 formulate problems and select or design solutions to them
- 3.2.4 select and apply appropriate techniques and principles to develop data structures and algorithms for the solution of problems
- 3.2.5 implement data structures and algorithms to produce effective, documented and tested systems, using existing or purpose designed software and general special purpose hardware as appropriate

3.3 Practical Skill

Learners should be able to:

- 3.3.1 design and develop a solution to a given problem using Visual Basic (6.0 or .Net)

4.0 ASSESSMENT

4.1 Skills Weighting

Paper 1

Skill	Assessment objective	Weighting (%)
Knowledge and understanding	3.1	60
Problem Solving	3.2	40
TOTAL		100

Paper 2

Skill	Assessment objective	Weighting (%)
Knowledge and understanding	3.1	30
Problem solving	3.2	60
Practical skills	3.3	10
TOTAL		100

4.2 Scheme Of Assessment

The subject will be examined in three papers.

Paper	Paper description	Duration	Marks	Weighting (%)
1	Structured theory paper on Sections 1, 3, 4, 6 and 7 of the Syllabus. The paper consists of 10 - 12 compulsory questions.	3 hours	100	50
2	Written practical paper with two sections. Section A is on Section 5 of the syllabus with 5 compulsory questions. Section B is on Sections 2, 3 and 4 of the Syllabus and candidates are supposed to make a choice of two (2) out of three (3) questions.	3 hours	100	25
3	Project as specified in the project guidelines	5 terms	100	25
TOTAL				100

5.0 GRADE DESCRIPTORS

The scheme of assessment is intended to encourage positive achievement by all learners. Grade descriptors are therefore provided for key grades A, B and C to give a general indication of the standards of achievement expected of learners awarded particular grades. The descriptors must be interpreted in relation to the content specified by the Computing syllabus but are not designed to define that content. The grade awarded will depend in practice on the extent to which the learner has met the overall assessment objectives.

Grade	Descriptors
A	<ol style="list-style-type: none"> 1. Analyse problems and procedures in application areas 2. Evaluate situations and come up with distinguished solutions using system development tools 3. Apply ERDs to solve given problems and interpret the ERDs inclusive of cardinalities in the description 4. Analyse schemas 5. Construct binary trees and arrays 6. Calculate binary numbers up to errors 7. Use characteristics of VB programming in the code (inheritance, polymorphism and encapsulation) 8. Present the usability code with annotations, comments and error handling techniques 9. Solve problems using searching and sorting techniques 10. Solve problems using logic gates 11. Justify the choice of access methods
B	<ol style="list-style-type: none"> 1. Interpret problems and procedures of computer application areas 2. Assess situations and come up with solutions using system development tools 3. Draw and label ERDs and explain the flow of data in the system 4. Explain schemas 5. Interpret binary trees and arrays 6. Calculate binary numbers up to floating point 7. Explain characteristics of VB(inheritance, polymorphism and encapsulation) 8. Code but without the element of error handling 9. Explain searching and sorting techniques 10. Draw and explain logic gates 11. Explain access methods

Grade	Descriptors
C	<ol style="list-style-type: none">1. Identify problems and procedures of computer application areas2. Explain situations using system development tools3. Draw and label ERDs to solve given problems4. Identify schemas5. Convert binary numbers6. Distinguish between static and dynamic data structures7. State characteristics of VB(inheritance, polymorphism and encapsulation)8. Code but with some bugs (errors) in the code9. Identify searching and sorting techniques10. Draw logic gates11. Identify access methods

6.0 METHODOLOGY AND TIME ALLOCATION

6.1 Methodology

A learner-centred approach is essential. Some of the participatory and learner-centred methods that may be used to teach Computing include:

- demonstration and observations
- discussions
- directed practice
- question and answer
- case studies
- group work
- discovery and experimentation
- research and presentations
- field trips
- presentation by resource person
- project work
- visual displays
- use of audio-visual media
- seminars
- collaborative pair work

6.2 Time allocation

At least **10** periods per week per class should be allocated to computer lessons. (A period is at least 35 – 40 minutes)

7.0 PRESENTATION OF CONTENT

The syllabus consists of seven sections. These are:

- 7.1 Applications of computers and their social and economic implications
- 7.2 Systems analysis and design
- 7.3 File organisation and database concepts
- 7.4 Data structure types and representations
- 7.5 Algorithm design and programming concepts
- 7.6 Software and hardware
- 7.7 Architecture, data communications and networks

7.1 SECTION 1: APPLICATIONS OF COMPUTERS AND THEIR SOCIAL AND ECONOMIC IMPLICATIONS

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
<p>7.1.1 The range and scope of computer applications</p> <ul style="list-style-type: none"> • General application areas 	<ul style="list-style-type: none"> • identify general application areas using examples • explain a variety of computer applications • state purposes, requirements and designs of the applications 	<ul style="list-style-type: none"> • Communication and information systems <ul style="list-style-type: none"> - VOIP - fax - email - e-conferencing - database systems - social networks - other internet services • Commercial and general data processing <ul style="list-style-type: none"> - e-banking - pos - stock control - e-commerce - record management systems, for example patient records management systems • Industrial, technical and scientific uses <ul style="list-style-type: none"> - weather forecasting - CAD/CAM - simulation and modelling, for example, flight simulation - image processing, for example, GPRs • Monitoring and control systems <ul style="list-style-type: none"> - traffic control - nuclear power station - chemical process control - monitoring patients • Automation, embedded and robotics <ul style="list-style-type: none"> - domestic equipment - automatic navigation systems - automated fuel injection systems - industrial robots 	<ul style="list-style-type: none"> • Researching and reporting on general application areas of computers • Discussing variety of computer applications • Discussing purposes and requirements of applications • Conducting field trips to identify general computer application areas

Topic	Objectives Learners should be able to:	Content	Suggested Learning Activities And Notes
		<ul style="list-style-type: none"> • Expert systems and artificial intelligence <ul style="list-style-type: none"> - mineral prospecting - medical diagnosis - speech recognition • Entertainment, education and training <ul style="list-style-type: none"> - e-learning - e-marking - e-registration - multimedia - multimedia systems - animation for television and film 	
7.1.2 Social and economic implications of the use of computers	<ul style="list-style-type: none"> • determine the social and economic implications of the use of computers on people and organisations • list the advantages and disadvantages of the use of computers across a range of application areas • justify the need for privacy and data integrity 	<ul style="list-style-type: none"> • Economic and social implications of the use of computers • Privacy and data integrity <ul style="list-style-type: none"> - reliability - security - flexibility of systems - measures to combat computer crime - computer crime - hacking and viruses - data protection legislation - computer ethics • Advantages and disadvantages of the use of computers across a range of application areas 	<ul style="list-style-type: none"> • Researching on the social and economic implications of the use of computers on people and organisations • Identifying new products and services brought about by the use of computers • Discussing the effects of the use of computers across a range of application areas • Investigating changes in employment and retraining • Debating on the need of privacy and data integrity • Discussing security measures

7.2 SECTION 2: SYSTEMS ANALYSIS AND DESIGN

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.2.1 Types of systems	<ul style="list-style-type: none"> • explain different types of systems • describe what is needed to support the systems • explain the most suitable system for a given application 	<ul style="list-style-type: none"> • Types of systems <ul style="list-style-type: none"> - Batch processing - On-line - Distributed and centralised - Control - Automated and embedded - Multimedia 	<ul style="list-style-type: none"> • Visiting and reporting on various industries to examine different types of systems • Discussing on types of systems • Describing what is needed to support the systems • Discussing the most suitable system for a given application
7.2.2 Systems development life cycle (SDLC)	<ul style="list-style-type: none"> • identify the stages of the generic SDLC • outline the activities at each stage of the generic SDLC • explain the importance of SDLC stages • design inputs, outputs and processing needs of a system using diagrammatic representation where appropriate including the use of Data Flow Diagrams (DFDs) and system flowcharts 	<ul style="list-style-type: none"> • Generic SDLC stages <ul style="list-style-type: none"> - feasibility study - data collection - analysis of a problem - system design - system development and testing - system implementation - system maintenance • Importance of SDLC stages • Design aspects <ul style="list-style-type: none"> - output formats - input formats - system processing - DFDs - program design <ul style="list-style-type: none"> ▪ system flowcharts ▪ program flowcharts ▪ pseudo-codes 	<ul style="list-style-type: none"> • Researching and reporting on the stages of the SDLC and activities at each stage NB: learners can do a case study • Discussing the importance of each stage • Designing inputs formats, outputs formats and processing • Constructing DFDs, system flowcharts, program flowcharts and pseudo-codes

7.3 SECTION 3: FILE ORGANISATION AND DATABASE CONCEPTS

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.3.1 File organisation	<ul style="list-style-type: none"> • identify types of file organisation • explain types of file organisation • justify the most appropriate file organisation 	<ul style="list-style-type: none"> • Types of file organisation <ul style="list-style-type: none"> - serial - sequential - random/direct - index sequential 	<ul style="list-style-type: none"> • Researching and reporting on types of file organisation • Discussing application of file organisations <p>NB: include records and fields, variable length records, fixed length records</p>
7.3.2 File access methods	<ul style="list-style-type: none"> • identify types of access methods for a given application • explain types of access methods • justify the most appropriate file access method for a given application 	<ul style="list-style-type: none"> • File access <ul style="list-style-type: none"> - serial - sequential - random/direct - index sequential 	<ul style="list-style-type: none"> • Identifying types of access methods to use for a given file • Explaining types of access methods • Discussing the most appropriate file access method for a given application
7.3.3 File security and privacy	<ul style="list-style-type: none"> • explain file security measures • justify the appropriate measure for a given scenario 	<ul style="list-style-type: none"> • File security and privacy: <ul style="list-style-type: none"> - Backups and archives - Access rights - File generations - Transaction logs - Passwords and data encryption - Recovery procedures - Biometric procedures 	<ul style="list-style-type: none"> • Researching and presenting on various file security measures for a given application <p>NB: include disaster recovery plan</p>
7.3.4 Database systems	<ul style="list-style-type: none"> • define database terms • compare files and databases 	<ul style="list-style-type: none"> • Database terms • Comparison of files and database systems 	<ul style="list-style-type: none"> • Discussing database terms • Researching and presenting on files and databases

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.3.5 Database models	<ul style="list-style-type: none"> • explain database models 	<ul style="list-style-type: none"> • Database models: <ul style="list-style-type: none"> - Hierarchical - Relational - Network - Object oriented 	<ul style="list-style-type: none"> • Discussing database models
7.3.6 Entity Relationship Diagrams (ERDs)	<ul style="list-style-type: none"> • draw an ERD for a given scenario • interpret an ERD 	<ul style="list-style-type: none"> • entities • relations cardinality 	<ul style="list-style-type: none"> • Constructing an ERD for a given scenario • analysing an ERD
7.3.7 Data Base Management System (DBMS)	<ul style="list-style-type: none"> • outline the structure of a DBMS • describe database architecture • explain functions of DBMS 	<ul style="list-style-type: none"> • schemas <ul style="list-style-type: none"> - external - conceptual/logical - internal • DBMS <ul style="list-style-type: none"> - data dictionary - Data Description Language (DDL) - Data Manipulation Language (DML) - Structured Query Language (SQL) 	<ul style="list-style-type: none"> • Describing the structure of a DBMS • Discussing on database architecture • Discussing use of schemas
7.3.8 Database security	<ul style="list-style-type: none"> • explain the importance of securing a database • evaluate different methods of securing a database 	<ul style="list-style-type: none"> • access control <ul style="list-style-type: none"> - views - passwords • physical security • data encryption 	<ul style="list-style-type: none"> • Discussing the importance of securing a database • Researching and reporting on different methods of securing a database

7.4 SECTION 4: DATA STRUCTURES, TYPES AND REPRESENTATIONS

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.4.1 Data structures	<ul style="list-style-type: none"> • Distinguish between dynamic and static data structures • Construct operations on binary trees and arrays 	<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 	<ul style="list-style-type: none"> • Distinguishing between dynamic and static data structures • Constructing operations on binary trees and arrays
7.4.2 Data types and representations	<ul style="list-style-type: none"> • Identify data types • Outline units of storage • Convert number bases • Explain data representations codes • Calculate errors • Analyse errors 	<ul style="list-style-type: none"> • Data types <ul style="list-style-type: none"> - integer - Boolean - character - string - real • Units of storage <ul style="list-style-type: none"> - bit - byte - word - nibble • Number bases <ul style="list-style-type: none"> - binary numbers, - octal - hexadecimal, - floating point, - BCD, • Data representation <ul style="list-style-type: none"> - ASCII - BCD - EBC • Computer arithmetic errors <ul style="list-style-type: none"> - underflow - overflow 	<ul style="list-style-type: none"> • Discussing data types • Discussing units of storage • Researching and reporting on data representation codes – • Discussing errors • Debugging identified errors

7.5 SECTION 5: ALGORITHM DESIGN AND PROGRAMMING CONCEPTS

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.5.1 Standard algorithms	<ul style="list-style-type: none"> • Describe algorithms using pseudo codes and flowcharts • Apply searching and sorting to solve problems 	<ul style="list-style-type: none"> • Algorithm structures <ul style="list-style-type: none"> - selection - repetition/iteration - recursion - linear • Sorting algorithms <ul style="list-style-type: none"> - bubble sort - quick sort • Searching algorithms <ul style="list-style-type: none"> - linear - binary search 	<ul style="list-style-type: none"> • Designing algorithms using pseudo codes and flowcharts • Solving searching and sorting problems
7.5.2 System design approaches	<ul style="list-style-type: none"> • Compare top-down and bottom-up approaches to program design 	<ul style="list-style-type: none"> • System design approaches <ul style="list-style-type: none"> - top-down design - bottom-up design 	<ul style="list-style-type: none"> • Discussing top-down and bottom-up approaches to program design • Demonstrating the use of top-down and bottom-up approaches to program design

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.5.3 Programming languages	<ul style="list-style-type: none"> • explain the features of programming languages • describe the features of low level languages • describe the features of high level languages • compare high level and low level languages 	<ul style="list-style-type: none"> • Language features <ul style="list-style-type: none"> - programming constructs - constants - variables - expressions - statements - control structure - block structure - variables <ul style="list-style-type: none"> ▪ local ▪ global - functions and procedures - parameter passing <ul style="list-style-type: none"> ▪ by value ▪ by reference • input, output and file handling operations • Programming languages <ul style="list-style-type: none"> - High Level language (HLL) - Low Level language (LLL) • LLL types <ul style="list-style-type: none"> - machine language - assembly language • HLL types <ul style="list-style-type: none"> - imperative/procedural - declarative - general purpose - special purpose - Object Oriented Programming (OOP) 	<ul style="list-style-type: none"> • Discussing the features of programming languages • Describing characteristics or features of high level languages and their proper use • Analysing high level and low level languages

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.5.4 Visual Basic programming VB (6.0 or .net)	<ul style="list-style-type: none"> • define basic VB programming terms • identify data types • declare variables • use operators and VB syntax correctly • explain the scope of a variable • code simple VB programs 	<ul style="list-style-type: none"> • Basic structure of a VB program • Data types in VB <ul style="list-style-type: none"> - integer - real - character - string - word - Boolean • Variables • constants • Syntax <ul style="list-style-type: none"> - statements - expressions • Operators <ul style="list-style-type: none"> - arithmetic - logical - assignment - comparison • Variable scope <ul style="list-style-type: none"> - global - local • VB control interfaces <ul style="list-style-type: none"> - textbox - command button - check box - option button - list box • Control structures <ul style="list-style-type: none"> - sequence - repetition <ul style="list-style-type: none"> ▪ Do...While ▪ For... Next 	<ul style="list-style-type: none"> • Discussing basic VB programming terms • Explaining data types • Declaring variables correctly as they apply to VB programming • Choosing correct identifier names • Using correct operators and VB syntax • Designing VB interfaces using VB controls • Discussing the scope of variables • Coding VB programs such as simple arithmetic problems: addition, subtraction, division and multiplication of two or more variables, use of control structures • Programs that calculate (area, roots of a quadratic equation and grading of marks)

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
	<ul style="list-style-type: none"> • develop a VB code • identify types of errors • apply error handling techniques in a VB program • debug errors in programs 	<p>Do... Until</p> <ul style="list-style-type: none"> - selection/decision <ul style="list-style-type: none"> ▪ Select Case ▪ Elseif ▪ Case ...Of ▪ If...Then...Else ▪ Cascaded/Nested If • Errors in VB <ul style="list-style-type: none"> - types of errors <ul style="list-style-type: none"> ▪ syntax, ▪ logical, ▪ run time - error handling techniques <ul style="list-style-type: none"> ▪ try catch as exception 	<ul style="list-style-type: none"> • Running a VB code • Dry run a VB code • Applying error handling techniques in a VB program • Debugging errors in programs NB: annotate statement and correct indentation

7.6 SECTION 6: SOFTWARE & HARDWARE

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.6.1 Application software	<ul style="list-style-type: none"> • describe different application packages and their areas of application • explain the advantages and disadvantages of tailor made software and general purpose packages 	<ul style="list-style-type: none"> • General purpose packages <ul style="list-style-type: none"> - word-processing - spreadsheet - presentation - data management package - desktop publishing - information retrieval packages - data logging, CAD and authoring packages - tailor made software • Tailor made software and general purpose packages 	<ul style="list-style-type: none"> • Discussing different application packages • Researching and reporting on different application software within their locality • Discussing advantages and disadvantages of tailor made software and general purpose packages
7.6.2 Systems software	<ul style="list-style-type: none"> • describe systems software types • demonstrate knowledge of computer utilities and tools • identify language processors • explain characteristics of language processors 	<ul style="list-style-type: none"> • System software types • Utilities and tools <ul style="list-style-type: none"> - antivirus - disk formatting and defragmentation - file compression - registry cleaner - translator • Language processors and their characteristics <ul style="list-style-type: none"> - language translators <ul style="list-style-type: none"> ▪ compilers ▪ interpreters ▪ assembler - lexical analysis <ul style="list-style-type: none"> ▪ reverse polish form ▪ code generators ▪ run time support routines 	<ul style="list-style-type: none"> • Discussing types of systems software • Demonstrating knowledge of computer utilities and tools • Researching and reporting on language processors • Discussing characteristics of language processors

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.6.3	<ul style="list-style-type: none"> • outline functions of an Operating System (OS) • explain the characteristics of different types of OS • explain different types of user interfaces • choose a suitable user interface for a given application 	<ul style="list-style-type: none"> • OS functions • OS types <ul style="list-style-type: none"> - batch - real-time - single user - multi-user - multi-tasking - networking systems • Examples of OS <ul style="list-style-type: none"> - Linux - windows family - mobile OS <ul style="list-style-type: none"> ▪ Android ▪ Symbian • User interfaces <ul style="list-style-type: none"> - command line - GUI - job control language 	<ul style="list-style-type: none"> • Explaining the functions and features of different types of OS • Designing a suitable user interface for a given application
7.6.4 Hardware	<ul style="list-style-type: none"> • outline categories of peripherals • Explain advantages and disadvantages of various hardware devices • choose appropriate device for a given application • explain processor components 	<ul style="list-style-type: none"> • Peripheral devices <ul style="list-style-type: none"> - output - input • Storage <ul style="list-style-type: none"> -primary (ROM, RAM) -secondary • Processor components <ul style="list-style-type: none"> - CU - ALU - registers 	<ul style="list-style-type: none"> • Researching and reporting on different peripherals • Discussing characteristics of different devices • Discussing devices for applications • Describing the uses of processor components

7.7 SECTION 7: ARCHITECTURE, DATA COMMUNICATIONS AND NETWORKS

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.7.1 Architecture	<ul style="list-style-type: none"> • describe the Von Neumann and Harvard architecture • explain the use of buses • describe the fetch-decode-execute cycle • explain the importance of memory mapping • identify types of interrupts • justify why computers use interrupts • explain addressing modes • explain different types and uses of registers • construct a truth table of up to three inputs • design logic gates from an existing truth table • calculate the outcome from a set of logic gates given the inputs 	<ul style="list-style-type: none"> • Von Neumann architecture • Harvard architecture • Pipelining • buses <ul style="list-style-type: none"> - control - address - data • Fetch-decode-execute cycle • Memory mapping • Interrupts • Addressing modes <ul style="list-style-type: none"> - direct - indirect - immediate - indexed - relative • Registers • Logic gates <ul style="list-style-type: none"> - AND - OR - NAND - NOR - NOT - XOR - XNOR 	<ul style="list-style-type: none"> • Discussing on the Von Neumann and Harvard architecture • Discussing on the use of buses • Illustrating the fetch-decode-execute cycle • Describing the importance of memory mapping • Distinguishing types of interrupts • Discussing computer interrupts • Researching and reporting on addressing modes and use of registers • Constructing a truth table of up to three inputs • Designing logic gates from an existing table • Calculating the outcome from a set of logic gates given the inputs

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.7.2 Data transmission	<ul style="list-style-type: none"> • explain types of data transmission media • outline the advantages and disadvantages of data transmission media • explain data transmission mode • explain the need for protocols to establish communication • distinguish between circuit switching and packet switching 	<ul style="list-style-type: none"> • Transmission media types, advantages and disadvantages <ul style="list-style-type: none"> - cables <ul style="list-style-type: none"> ▪ UTP ▪ fibre optic ▪ coaxial - wireless <ul style="list-style-type: none"> ▪ Bluetooth ▪ radio ▪ WIFI ▪ WIMAX • Transmission modes <ul style="list-style-type: none"> - simplex - half duplex - full duplex • Protocols <ul style="list-style-type: none"> - OSI - TCP/IP • Circuit switching • Packet switching 	<ul style="list-style-type: none"> • Describing types of data transmission media and modes • Comparing and contrasting data transmission media • Configuring a UTP cable • Researching and reporting on protocols • Comparing circuit switching and packet switching • Configuring a computer to the network using TCP/IP <p>NB: Learners are not expected to have a detailed knowledge of specific protocol but defining and explaining their need in data transmission</p>

Topic	Objectives Learners should be able to:	Content	Suggested learning activities and notes
7.7.3 Networking	<ul style="list-style-type: none"> • describe the characteristics of networks • outline the merits and demerits of computer networks • illustrate network topologies • describe the operations of network devices • distinguish network device types and characteristics • describe media access methods 	<ul style="list-style-type: none"> • Types of networks <ul style="list-style-type: none"> - LAN - WAN - MAN • Network topologies <ul style="list-style-type: none"> - star - ring - bus - mesh - hybrid • Network device types and characteristics <ul style="list-style-type: none"> - hub - switch - router - gateway - bridge • Media access <ul style="list-style-type: none"> - CSMA/CD - CSMA/CA - token passing - contention 	<ul style="list-style-type: none"> • Discussing the characteristics of networks • Discussing the merits and demerits of computer networks • Researching and reporting on the operations of network devices • Constructing network topologies • Differentiating network devices • Discussing media access methods

APPENDIX I: ACRONYMS

ALU	Arithmetic Logic Unit
ASCII	American Standard Code for Information Interchange
BCD	Binary Coded Decimal
CAD	Computer Aided Design
CAM	Computer Aided Manufacture
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
CU	Control Unit
DBMS	Database Management System
DDL	Data Description Language
DFD	Data Flow Diagram
DML	Data Manipulation Language
EBCDIC	Extended Binary Coded Decimal Interchange Code
GPRS	Global Positioning Radio Systems
ERD	Entity Relationship Diagram
HLL	High Level Language
LLL	Low Level Language
LAN	Local Area Network
MAN	Metropolitan Area Network
NAND	NOT AND
OOP	Object Oriented Programming
OS	Operating Systems
OSI	Open System Interconnection
POS	Point of Sale
POST	Point of Sale on Terminal
RAM	Random Access Memory
ROM	Read Only Memory
SDLC	System Development Life Cycle
TCP/IP	Transmission Control Protocol/Internet Protocol
UTP	Universal Transfer Protocol
XOR	Exclusive OR
XNOR	Exclusive NOR
VB	Visual Basic
VoIP	Voice Over Internet Protocol
WAN	Wide Area Network

WiMax	World Wide Inter-operability for Marking Access
WiFi	Wireless Fidelity

APPENDIX II: PROJECT GUIDELINES/PRACTICAL

The project must not exceed 50 pages excluding appendices.

The project must include the following layout:

- I. Cover page
- II. Table of contents
- III. 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 and marked by the teacher. The project and the marks must be submitted to ZIMSEC at the same time as the scripts for the last theory paper to be written. A copy of the marks should be retained at the school. Normal channels of packaging and postage are to be followed.

SECTION A

Selection, investigation and analysis

- Define a problem
 - Choice of problem area and background analysis.
 - Investigation of the current system
 - Research instruments e.g. questionnaire, record inspection, interviews and observation.
 - Identify problems with the current system. [5]
 - Requirements specification
 - User
 - Software
 - Hardware. [5]
 - Aims and objectives. [5]
 - Evidence that the research has been carried out. [5]
 - Examples are filled in questionnaires, interviews with respondent answers, sample documents and write up on observation.
- TOTAL** [5]
25

SECTION B

Design

- Consideration of alternative methods. [3]
 - Justification of method of solution [2]
- Input design [4]
 - Appropriate data capture forms and screen layouts [4]
- Data structures/File design [5]
- Overall plan [3]
- Output design [5]
 - Specification and design of the required output
 - Interface design (on screen commands)

- Test strategy/Test plan [3]
 - design and document a test plan that includes
 - test data and expected outcomes

TOTAL

25

SECTION C

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 [7]
 - Algorithms
 - pseudo codes
 - flowcharts for modules
 - codes/program listings

- User documentation
 - Installation
 - Running the system
 - Navigation of the system
 - Exiting the system
- [8]
- TOTAL** **25**

SECTION D

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 of the system
 - Extent of success in meeting the system objectives as stated in the system requirements specification
 - Achievements
 - Limitations
 - Evaluate results against the system objectives – achievements and limitations
 - Opportunities for future developments

[5]
- TOTAL** **15**

SECTION E

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 work carefully organised? 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]

- TOTAL 10**
- GRAND TOTAL 100**

APPENDIX III: MINIMUM RESOURCES AND EQUIPMENT REQUIREMENTS

The following are the minimum requirements. Centres must ensure that their equipment and facilities are adequate for learners to be able to satisfy the requirements of the syllabus. The hardware facilities needed must depend on the number of learners and should be sufficient for all learners to become familiar with system, software and application packages; and to develop their own software.

- The learner computer ratio should be one student per computer during lesson time. This does not necessarily follow that if a school has 50 learners and 20 computers, then they cannot offer the subject. Learners can still be divided into groups that ensure 1:1 ratio but caution must be taken to ensure that they complete the mandatory ten 40-minute periods per week
- Internet connectivity is recommended for research purposes

The table below shows the recommended computer specifications:

DESCRIPTION	SPECIFICATION
RAM	1GB
Hard Drive	80GB
Processor Speed	2.4GHz
Operating System	Windows /Linux
Programming Language	Visual Basic 6.0/VB.net
Application Software	MS Office
Printer	Laser Printer
Local Area Network	Wired/wireless
Data Projector	
Scanner	Flat bed

APPENDIX IV: POSSIBLE CAREER OPPORTUNITIES

1. Learners who attain this qualification may qualify for entry into different universities or colleges.

Possible career paths or programs to be pursued	Possible subject combinations
<ul style="list-style-type: none">• Computer science• Software engineering• Computer security• Information Technology• Telecommunications	Mathematics, Computing and any other science subject
<ul style="list-style-type: none">• Information systems• e-commerce	Computing and any other commercial subjects
<ul style="list-style-type: none">• B.Ed Computers (Teacher education)	Computing and any other combination of subjects

NB: Teachers to constantly remind learners of career opportunities throughout the course.

2. Learners may also pursue apprenticeship opportunities in the fields of IT in such companies as TelOne, ZESA and NRZ or pursue entry level jobs in industry such as data capturing.

APPENDIX V: REFERENCE BOOKS AND JOURNALS

- Bradley, R, **Understanding Computer Science for Advanced Level Fourth Edition**, Stanley Thorne; 0748719792
- **British Computer Society Glossary of Computing Terms (9th Edition)**, Longman; 0-582-36967-3
- Crawford. R (2010), **ICT**, Pearson Education Limited
- Heathcote P. M and Bond. K, **Computing A-Level Study Guide**. Letts; 1857586018
- Heathcote P. M and Langfield S, **‘A’ Level Computing Fifth Edition**, 2004, Payne Galway Publishers
- Knott G and Waites, **Computing Business** Education Publishers; 0-907679-87-0
- Kalicharan N, **Computer Studies**
- Educational Websites
- British Computer Society (2005), **The BCS Glossary ICT and Computer Terms**, McMillan, UK
- Brown, G and D Watson (2010), **IGCSE ICT**, Hodder Education, UK
- Doyle, S. (2011), **Information Systems for you 4th Edition**, Nelson Thompson, UK
- French, C, S (1996), **Data Processing and ICT 5th edition**, Thompson, UK
- Roderick, T & Rushbrook, G (2002), **ICT for GCSE**, Oxford University Press, UK
- Taylor, G. (1991), **GCSE Computer Studies and Information Technology**, McMillan, UK