Republic of Namibia

MINISTRY OF EDUCATION, ARTS AND CULTURE

NAMIBIA SENIOR SECONDARY CERTIFICATE (NSSC)

COMPUTER SCIENCE SYLLABUS

ADVANCED SUBSIDIARY LEVEL

SYLLABUS CODE: 8231

GRADE 12

FOR IMPLEMENTATION IN 2021

FOR FIRST EXAMINATION IN 2021

Ministry of Education, Arts and Culture
National Institute for Educational Development (NIED)
Private Bag 2034
Okahandja
Namibia

Computer Science Syllabus Advanced Subsidiary level, Grade 12

ISBN: 978-99945-2-219-4

Printed by NIED

Publication date: 2019

TABLE OF CONTENTS

1. INTRODUCTION

The Namibia Senior Secondary Certificate Advanced Subsidiary (NSSCAS) level is designed as a one year course for examination after the successful completion of the Namibia Senior Secondary Certificate for Ordinary (NSSCO) level. The syllabus is designed to meet the requirements of the National Curriculum for Basic Education and has been approved by the National Examination, Assessment and Certification Board (NEACB).

The National Curriculum Guidelines, applicable at the stage of Senior Secondary Education (Grades 10-12) and at equivalent stages of non-formal education, as a part of life-long learning, recognise the uniqueness of the learner and adhere to the philosophy of learner centred education.

The Namibia National Curriculum Guidelines:

- recognise that learning involves developing values and attitudes as well as knowledge and skills
- promote self-awareness and an understanding of the attitudes, values and beliefs of others in a multilingual and multicultural society
- encourage respect for human rights and freedom of speech
- provide insight and understanding of crucial "global" issues in a rapidly changing world which affects quality of life: the AIDS pandemic, global warming, environmental degradation, distribution of wealth, expanding and increasing conflicts, the technological explosion and increased connectivity
- recognise that as information in its various forms becomes more accessible, learners need to develop higher cognitive skills of analysis, interpretation and evaluation to use information effectively
- seek to challenge and to motivate learners to reach their full potential and to contribute positively to the environment, economy and society

Thus the Namibia National Curriculum Guidelines provide opportunities for developing essential key skills across the various fields of study. Such skills cannot be developed in isolation and they may differ from context to context according to a field of study. The skills marked with an * are relevant to this syllabus.

The skills are:

- communication skills *
- numeracy skills *
- information skills *
- problem-solving skills *
- self-management and competitive skills *
- social and cooperative skills *
- physical skills
- work and study skills *
- critical and creative thinking*

2. RATIONALE

Computer Science is a dynamic, living and cultural product. It is more than an accumulation of facts, skills and knowledge. The learning of Computer Science involves conceptual structures, strategies of problem solving and attitudes towards and appreciation of technology. Increasingly in the modern world, acquisition of computer skills is becoming necessary for employment, educational development and leisure. The Computer Science course intends to furnish learners with in-depth knowledge and specialised skills in programming and specific knowledge of the internal operations of a computer.

The senior secondary curriculum for Computer Science strives to prepare learners to function effectively in the 21st century by providing a basis to utilise the skills and knowledge of computing to:

- acquire a specific understanding and perspective of the use of computer systems in order to be productive in a society increasingly dependent on technology
- acquire specific skills of computer programming
- acquire the necessary skills and knowledge to seek employment in areas that utilise computing
- continue developing their knowledge and understanding of computing for entry into higher education for seeking specialisation in chosen computing fields
- use information responsibly and ethically

3. AIMS

The aims of the syllabus are the same for all learners. These are set out below and describe the educational purposes of a course in Computer Science for the NSSCAS examination. They are not listed in order of priority.

The aims are to enable learners to:

1. develop an understanding of the principles of computing
2. develop an in-depth knowledge to design and program user-friendly computer-based solutions for problems
3. critically analyse the impact of computer usage on social, economic and ethical issues
4. use information systems to access various resources all over the world for learning purposes
5. develop computational thinking skills
6. make good decisions regarding information and programming
7. effectively use computer technologies to:
    - develop problem-solving skills which lead to logical problem-solving
    - acquire effective communication skills
    - gain knowledge of using technology responsibly
    - use programming skills to create and develop effective computer software that is efficient, precise and accurate

4. ADDITIONAL INFORMATION

4.1 Guided learning hours

The NSSCAS level syllabuses are designed on the assumption that learners have about 180 guided learning hours per subject over the duration of the course (1 year), but this is for guidance only. The number of hours required to gain the qualification may vary according to local conditions and the learners' prior experience of the subject. *The National Curriculum for Basic Education (NCBE)* indicates that this subject will be taught for 9 periods of 40 minutes each per 7-day cycle, over a year.

4.2 Recommended prior learning

It is required that learners who are beginning this course should have successfully completed the NSSCO course in Computer Studies.

4.3 Progression

NSSCAS Computer Science provides a suitable foundation for the study of Computer Science related courses in higher education. Depending on the local university entrance requirement, it may permit or assist progression directly to university courses in Computer Science or other related course in higher education. Equally it is suitable for learners intending to pursue careers or further study in software development, systems analysis, hardware solutions, etc.

4.4 Grading and reporting

NSSCAS results are shown by one of the grades a, b, c, d or e indicating the standard achieved, grade a being the highest and grade e the lowest. 'Ungraded' indicates that the candidate has failed to reach the standard required for a pass at NSSCAS level.

4.5 Support materials and approved textbooks

Copies of NSSCAS syllabuses, recent specimen material, question papers and examiner reports are sent to all schools. Assessment manuals in subjects, where applicable, are sent to schools. Approved learning support materials are available on the Senior Secondary Textbook Catalogue for Schools. The Textbook Catalogue is available on www.nied.edu.na

5. OVERVIEW OF LEARNING CONTENT

The learning content outlined below is designed to provide guidance to teachers as to what will be assessed in the overall evaluation of learners. They are not meant to limit, in any way, the teaching program of any particular school.

The syllabus covers the following **themes**:

Theme 1: Hardware, software and communication
Theme 2: Data representation
Theme 3: Circuits and programming
Theme 4: Security, privacy and data integrity

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 1: HARDWARE, SOFTWARE AND COMMUNICATION		
1.1 Types of hardware **1.1.1 Input, output and storage devices**	• understand the purpose, use and internal operation of input, output and storage devices	• describe the data flow of the following - two dimensional (2D) and three dimensional (3D) scanners - code readers - keyboards - mice and other pointing devices - digital cameras - display devices - inkjet, laser and 3D printers - 2D and 3D cutters - microphones, speakers and headphones - sensors and actuators - projectors [liquid-crystal display (LCD), light-emitting diodes (LED) and Digital Light Processing (DLP)] - interactive screens and whiteboards - optical storage devices - magnetic storage devices - solid state storage devices - Refreshable Braille Display (RBD) - virtual reality (VR) devices • describe uses of the above in a variety of everyday situations, e.g. - sensors in security devices, household appliances and control systems - Quick Response (QR) code readers for expanded information in a variety of formats - appropriate choices of storage devices

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 1: HARDWARE, SOFTWARE AND COMMUNICATION (CONTINUED)		
1.1.2 Processor and primary storage components	• understand the use of the main components of a processor	• identify the main components of a processor based on the Von Neumann model • explain the function and the purpose of the following individual parts of the processor: - control unit (CU) - arithmetic logic unit (ALU) - system clock - registers such as Current Instruction Register (CIR), Program Counter (PC), Memory Address Register (MAR), - Memory Data Register (MDR), Accumulator (ACU), Index Register, Status Register - buses (control, data, address) • describe how the bus width and clock speed affect a computer's performance
	• understand the fetch-execute cycle when a computer performs tasks	• describe the fetch/decode/execute/reset cycle and the effects of these stages of the cycle on specific registers • describe how interrupts are handled • explain modes of addressing - immediate - direct - indirect - indexed - relative
	• demonstrate their understanding of different types of processor	• explain the need for different types of processors - single core - multi core

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 1: HARDWARE, SOFTWARE AND COMMUNICATION (CONTINUED)		
1.1.2 Processor and primary storage components (continued)	• understand the main primary memories	• identify the main primary memories (ROM and RAM) • discuss factors that may influence the performance of Random-access memory (RAM) in terms of latency • identify different types of RAM - dynamic random-access memory (DRAM) - static random-access memory (SRAM) - synchronous dynamic random-access memory (SDRAM)
	• understand memory management	• explain the role of cache memory • explain the use of virtual memory, including the use of paging and segmentation
1.1.3 e-communication devices	• demonstrate their understanding of the use of devices used to enable computers to communicate (locally and remotely)	• explain the purpose of the following hardware (wireless or cabled) used to enable computers to communicate such as - NIC (Network Interface Card, including wireless) - switch - bridge - router - repeater - antenna / aerial - modem - gateway - satellite - Universal Serial Bus (USB) Wi-Fi adapter - wireless access point (WAP)
	• demonstrate understanding of network security	• explain the purpose of a - firewall - proxy server

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 1: HARDWARE, SOFTWARE AND COMMUNICATION (CONTINUED)		
1.2 Types of software **1.2.1 General application software**	• demonstrate general understanding of the design of databases	• justify the choice of using a relational database to overcome the limitations of using a flat file • modify an existing relational database in terms of data maintenance (use SQL) - insert into - delete from - update • write SQL script to query a database • apply the techniques of normalisation up to third normal form • define and evaluate a database management system (DBMS) • show understanding of terminology associated with databases, e.g. primary key, foreign key, secondary key, relationship, candidate key, attribute, tuple, entity, referential integrity • describe a DBMS in terms of - data management, including maintaining a data dictionary - data modelling - logical schema - data integrity - data security, including backup procedures and the use of access rights to individuals/groups of users • describe how and when the following software tools found within a DBMS are used - developer interface - query processor - debugger - SQL • describe how high-level languages can be used to access data stored in a database

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 1: HARDWARE, SOFTWARE AND COMMUNICATION (CONTINUED)		
1.2.2 Levels of languages, language translators and programming languages software	• understand different levels and types of languages	• discuss the levels of languages in terms of their relative merits and drawbacks - high-level languages - low-level languages • describe the features and give examples of imperative programming languages - procedural - event-driven - object-oriented programming system (OOPS) ▪ classes ▪ objects ▪ methods • describe the features and give examples of declarative programming languages - logic programming - functional programming • distinguish between imperative and declarative high-level programming languages

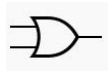
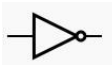
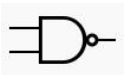
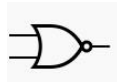
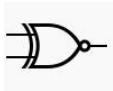
THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 1: HARDWARE, SOFTWARE AND COMMUNICATION (CONTINUED)		
1.2.3 System software and communication	• be familiar with operating systems	• evaluate and propose best solutions regarding - transaction processing - network operating systems (servers) - process-control operating system • explain the purpose of the following types of utility software - malware and spam filters - firewalls - disk repair - disk clean-up - backup and recovery software - disk partitioning
1.3 Software strategies	• understand various systematic approaches for problem solving	• represent software requirement specifications as algorithms using any of the following methods - structure diagrams - pseudocode - flowcharts

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 1: HARDWARE, SOFTWARE AND COMMUNICATION (CONTINUED)		
1.4 Networking	• demonstrate their understanding of protocols	• describe how communication links are established between computers by the use of protocols such as - Hyper Text Transfer Protocol (HTTP and HTTPS) - Transmission Control Protocol (TCP) / Internet Protocol (IP) (IPv4 and IPv6) - packet and circuit switching - Simple Mail Transfer Protocol (SMTP) - Post Office Protocol 3 (POP3) - File Transfer Protocol (FTP)
	• demonstrate their understanding of the Internet	• distinguish between the Internet and the World Wide Web (WWW)
	• understand the transfer of data using networks	• show understanding of - bit streaming (real time and on-demand) - speed (baud rate, bit rate, broadband) - Ethernet, including the use of collision detection and avoidance [Carrier Sense Multiple Access / Collision Detection (CSMA/CD)]
	• demonstrate their understanding of cloud computing	• analyse and evaluate cloud computing, including - software as a Service (SaaS) - platform as a Service (PaaS) - infrastructure as a Service (IaaS) - virtual storage - economic impact

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 2: DATA REPRESENTATION		
2.1 Data types, data structures and file handling	• be familiar with different data types	• use the following data types when designing programs - Integer - Real / Float - Char - Boolean - Date / Time - String / Text - Object
	• demonstrate their understanding of different models of data structures	• describe the following data structures used to solve problems and use diagrams or algorithms to explain how to initialise the structure, sort (basic), search, add, read and delete data in each type - arrays (single and two-dimensional) - files - linked lists • understand abstract data types - stacks - queues • analyse, evaluate and explain the choice of data types/data structures for a given problem, including the use of static and dynamic data structures
	• demonstrate their understanding of different models of file organisation	• explain the different types of file organisations - serial - sequential and indexed sequential - random / direct access

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 2: DATA REPRESENTATION (CONTINUED)		
2.1 Data types, data structures and file handling (continued)	• understand ways in which data is managed	• explain file generations • describe steps taken when data is added into or deleted from the different types of file organisation • describe how indexed sequential files may be implemented using indexing • describe how random access files may be implemented using hashing algorithms
2.2 Data representation	• understand binary systems	• convert positive or negative denary integers into binary and positive or negative binary integers into denary (a maximum of 16 bits will be used) • use binary in computer registers for a given application (such as in robotics, digital instruments and counting systems) • convert ASCII and Unicode characters into binary

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 2: DATA REPRESENTATION (CONTINUED)		
2.2 Data representation (continued)	• understand Binary Coded Decimal (BCD) • understand hexadecimal data systems	• convert a denary number to BCD and vice versa • describe practical applications where BCD is used • convert positive hexadecimal integers to and from denary (a maximum of four hexadecimal digits will be required) • convert positive hexadecimal integers to and from binary (a maximum of 16 bit binary numbers will be required) • represent numbers stored in registers and main memory as hexadecimal • explain the reasons for choosing hexadecimal notation to represent numbers • identify current uses of hexadecimal numbers in computing, such as - defining colours in HTML (Hypertext Markup Language) - Media Access Control (MAC) addresses - IPv6 addresses - assembly languages - debugging
2.3 Data storage	• understand the need for and principles of a variety of data storage formats (e.g. MIDI, JPEG, Bitmap, MP3, MP4)	• describe how data for bitmapped images or vector graphics is encoded • describe how sound is stored, represented and encoded • explain sampling, sampling rate and sampling resolution • describe the principles of data compression (lossless and lossy) applied to music/video, photos and text files

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 3: CIRCUITS AND PROGRAMMING		
3.1 Logic gates and circuits	• understand the function of basic logic gates	• correctly interpret and use AND, OR, NOT, NAND, NOR, XOR and XNOR gates AND  OR  NOT  NAND  NOR  XOR  XNOR 

<table>
<tr><th>THEMES AND TOPICS</th><th>GENERAL OBJECTIVES

Learners will:</th><th>SPECIFIC OBJECTIVES

Learners should be able to:</th></tr>
<tr><td colspan="3">THEME 3: CIRCUITS AND PROGRAMMING (CONTINUED)</td></tr>
<tr><td>3.1 Logic gates and circuits (continued)</td><td>• understand the function of basic logic gates (continued)</td><td>• produce truth tables for given logic circuits (maximum of 3 inputs per circuit, 2 inputs per gate and 6 gates)

<table><tr><th>x</th><th>y</th><th>z</th><th>output</th></tr><tr><td>0</td><td>0</td><td>0</td><td></td></tr><tr><td>0</td><td>0</td><td>1</td><td></td></tr><tr><td>0</td><td>1</td><td>0</td><td></td></tr><tr><td>0</td><td>1</td><td>1</td><td></td></tr><tr><td>1</td><td>0</td><td>0</td><td></td></tr><tr><td>1</td><td>0</td><td>1</td><td></td></tr><tr><td>1</td><td>1</td><td>0</td><td></td></tr><tr><td>1</td><td>1</td><td>1</td><td></td></tr></table>
• design a logic circuit and/or truth table from either
- a problem statement e.g. if x AND y are on OR z is on then the light will be on; or
- a logic expression e.g. F=(A+B)•(C+D)
• by inspecting a truth table, simplify a logic circuit by using fewer gates to solve the problem
• explain the structure of an integrated circuit
• describe how logic gates are used in a computer</td></tr>
<tr><td>3.2 Algorithms</td><td>• understand existing (third party) algorithms</td><td>• perform a dry run on an algorithm to
- determine the output/results
- correct errors
- modify designs
• use a trace table to show the results</td></tr>
</table>

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 3: CIRCUITS AND PROGRAMMING (CONTINUED)		
3.3 Algorithm design and development	• demonstrate their knowledge of the use of algorithms based on given problems or derived from module specifications to solve problems in a structured way, using logic and reason	• develop algorithms with multiple procedures/modules using algorithm tools such as - pseudocode - flowcharts • design algorithms which relate clearly to the requirements of the system where the problem has been observed, including stacks, queues and linked lists • define the scope of separate modules • use the following algorithmic constructs of sequence, selection, and repetition to solve problems - decision structures (if [including nested ifs], case) - loops or iteration for conditional and unconditional structures, including nested loops (pre-condition, e.g. while; post-condition, e.g. repeat; and counted, e.g. for) - data structures (arrays) • apply mathematical operators +, -, *, /, ^, MOD and DIV appropriately to solve problems • apply logical operators =, < >, <, >, <=, >= appropriately to solve problems • apply logic structures such as AND, OR and NOT
	• understand test strategies to analyse and evaluate algorithms	• select appropriate test data and compare actual test results to expected results • use a trace table to dry run an algorithm containing more than one loop/iteration and more than one decision structure • identify and rectify errors in an algorithm

THEMES AND TOPICS	GENERAL OBJECTIVES Learners will:	SPECIFIC OBJECTIVES Learners should be able to:
THEME 3: CIRCUITS AND PROGRAMMING (CONTINUED)		
3.3 Algorithm design and development (continued)	• understand the use of structure charts	• describe the purpose of a structure chart • create a structure chart for a given problem • use a structure chart to express the parameters passed between the various modules/procedures/functions which are part of the algorithm design • write an equivalent algorithm from a structure chart
3.4 Translate algorithms into computer programs*	• understand strategies to select appropriate implementation tools to solve problems	• write a program from a given design in a high-level language • update or modify an already written program
	• understand that algorithms are the design of solutions to problems	• apply the following - use of comments in program codes (annotation) - meaningful identifier names (variables, constants, programs, functions, etc.) - duplication/modification of existing code/procedures (reusability of modules) - program indentation
	• demonstrate their knowledge of organisation standards to develop and maintain programs	• describe and apply organisation standards in developing and maintaining programs - programming style - conventions for naming identifiers - annotation - hardware use - management of program versions

* Learners should use one of the following programming languages: Java, VB, Python or C++ or pseudocode

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 3: CIRCUITS AND PROGRAMMING (CONTINUED)		
3.4 Translate algorithms into computer programs (continued)	• understand facilities offered by the IDE (Integrated Development Environment)	• use the following facilities offered by the IDE - code editing - debugging tools - libraries
	• understand and apply test strategies to evaluate modules and programs developed	• describe and give examples of the following types of programming errors - syntax error - logic error - runtime error • select and apply suitable test data for the developed programs • use debugging tools to identify errors within programs • use module integration testing to identify errors caused by the interface between modules
	• understand the role of translators	• differentiate between compilers, interpreters and assemblers • explain that not all high-level languages use compilers or interpreters exclusively • explain the role of translators and use them
3.5 Utilise computer programs (code)	• be familiar with operations using computer programs	• apply searches - binary - serial

THEMES AND TOPICS	GENERAL OBJECTIVES *Learners will:*	SPECIFIC OBJECTIVES *Learners should be able to:*
THEME 4: SECURITY, PRIVACY AND DATA INTEGRITY		
4.1 Implications of information systems uses	• demonstrate their understanding of the impacts brought by the use of computer systems on our society	• apply the principles of the following when designing software solutions - security - privacy - data integrity • show understanding of methods used to prevent computer crimes/cybercrimes in both standalone computers and networks such as - development of complex security code systems - encryption of sensitive and confidential data ▪ hashing ▪ private and public keys ▪ cryptographic utility software - audit of access to the computer system (successful and not successful attempts) - security tokens ▪ virtual tokens and digital signatures ▪ hardware and software tokens - two step security • explain how errors are detected, as well as how they are corrected - data validation - data verification upon data entry - data verification during data transfer, including parity check, checksum, ARQ, echo back

6. ASSESSMENT OBJECTIVES

The three assessment objectives in NSSCAS Computer Science are:
A Knowledge with understanding
B Skills (Problem solving)
C Communication

A description of each assessment objective follows:

A. Knowledge with understanding

The learner should be able to:
1. describe and explain the importance of computing in a range and scope of information processing applications
2. analyse and reflect critically the impact of the use of computer technologies on social, legal, ethical and economic aspects
3. describe and explain the range of equipment, tools and techniques used to develop and implement high quality solutions to problems
4. describe and explain the functions of the main hardware and software components of information processing systems
5. use appropriate information systems terminology

B. Skills (Problem solving)

The learner should be able to:
1. analyse problems and draw up specifications for their computer-based solutions
2. design solutions by developing data structures and algorithms, in line with the specifications of the system
3. implement solutions using programming code (including annotations)
4. test, evaluate and refine solution systematically

C. Communication

The learner should be able to:
1. interpret and organise information
2. present information in a variety of forms
3. communicate information about computers, applications, problems and their solutions in appropriate ways

7. SCHEME OF ASSESSMENT

Assessment consists of two (2) written papers. All candidates must take both papers, and there is no choice of questions in either paper.

Differentiation

All learners are expected to pursue the same curriculum content and differentiation will be by outcome.

Description of papers

Paper 1	**3 hours**	**100 marks**	**60% weighting**

Paper 1 of the NSSCAS Computer Science examination will contain a variety of compulsory questions of variable mark value, which require answers of varying length and difficulty. The questions will be based on the whole syllabus in line with the assessment objectives. At least one question will require the construction of algorithms.

Paper 2	**2 hours 30 minutes**	**75 marks**	**40% weighting**

Paper 2 of the NSSCAS Computer Science is a compulsory paper consisting of questions set on theme 3 of the syllabus.

8. SPECIFICATION GRID

A rigid association between particular assessment objectives and individual examination components is not appropriate since any of the objectives can be assessed in any question. Nevertheless, the two components of the scheme will differ in the emphasis placed on the various objectives. For example, the assessment of the analysis of the impact of computer technologies uses (Objective 3: see assessment objectives) is particularly important in Paper 1, while the implementation of solutions (objective 8: see assessment objectives) is particularly important in Paper 2. The grid below is for general guidance only and illustrates where particular objectives might receive most emphasis.

Assessment objectives	**Paper 1 Marks**	**Paper 2 Marks**	**Total (%)**
A Knowledge and understanding	60	5	40
B Problem-solving and realisation	10	65	40
C Communication	30	5	20
TOTAL	**100**	**75**	**100**
Weighting	60%	40%	100%

In general, Paper 1 of the NSSCAS Computer Science examination will assess all themes, especially themes 1, 2 and 4. Paper 2 will concentrate on Theme 3 of the NSSCAS Computer Science syllabus. The assessment objectives will be interpreted as appropriate to the different topics in the four themes of the curriculum content.

9. GRADE DESCRIPTIONS

The scheme of assessment is intended to encourage positive achievement by all learners. Grade descriptions are therefore provided for judgmental grades a, c and e to give a general indication of the standards of achievement expected of learners awarded particular grades. The description must be interpreted in relation to the content specified by the Computer Science syllabus but are not designed to define that content. The grade awarded will depend in practice upon the extent to which the learner has met the assessment objective overall.

At Grade a, the learner is expected to demonstrate mastery in:

- knowing and understanding the principles and applications of computing
- understanding and formulating algorithms
- expressing ideas about the effects of computing on society and individuals
- applying methods and techniques to: – analyse a given problem
  - design a solution to a given problem
  - implement a solution to a given problem using an object oriented programming or procedural/event driven programming paradigms in a chosen programming language
- communicating clearly the solution to a given problem

At Grade c, the learner is expected to demonstrate general competence in:

- knowing and understanding the principles and applications of computing
- understanding and formulating algorithms
- expressing ideas about the effects of computing on society and individuals
- applying methods and techniques to: – analyse a given problem
  - design a solution to a given problem
  - implement a solution to a given problem using an object oriented programming or procedural/event driven programming paradigms in a chosen programming language
  - communicating generally the solution to a given problem

At Grade e, the learner is expected to demonstrate a basic competence in:

- knowing and understanding the principles and applications of computing
- understanding and formulating algorithms
- expressing ideas about the effects of computing on society and individuals
  - applying methods and techniques of:
    - analysing a given problem
    - designing a solution to a given problem
    - implementing a solution to a given problem using only query language and macro facilities
    - communicating generally the solution to a given problem

10. GLOSSARY OF TERMS

Array	A data structure consisting of a fixed number of elements all of the same type. An index is used to access each element of the array
Baud rate	the rate of change of signals per second sent by a communication device
Bitmap	a digital image composed of a matrix of dots
CSMA / CD	(Carrier Sense Multiple Access/Collision Detection) the transmission method used in Ethernet networks
Data integrity	ensuring that no unwanted changes occur to data during data entry or transmission
DBMS	(database management system) a collection of programs for creating and managing databases
DDR	(Double Data Rate) an advanced version of SDRAM that can transfer data twice as fast as regular SDRAM chips. This is because DDR memory can send and receive signals twice per clock cycle
DRAM	(dynamic random access memory) a type of memory that is typically used for data or program code that a computer processor needs to function
Float	variable with a fractional value
Hashing algorithm	a mathematical function that converts data of a variable size into a smaller, fixed-sized value
IaaS	(Infrastructure as a Service) a form of cloud computing that provides virtualised computing resources over the Internet – servers, storage and networking on demand
IDE	(Integrated Development Environment) an application that facilitates software development; a graphical user interface (GUI) is used to 'build' software, making use of a variety of available tools
IPv4	the fourth revision of the Internet Protocol and a widely used protocol in data communication over different kinds of networks
IPv6	the sixth revision and can support very large numbers of nodes, using 128-bit addresses
JPEG	(Joint Photographic Experts Group) a popular image file format commonly used by digital cameras to store photos since it supports 2^{24} or 16,777,216 colours. The compression algorithm is lossy, meaning some image quality is lost during the compression process
Lossless compression	a form of data compression algorithm that allows the reconstruction of the original data

Lossy compression	a data compression algorithm that cannot be reversed
Logical schema	a design-centric database structure that is independent of the database software used (the plan)
MAC	(Media Access Control) address is your computer's unique hardware number provided by the manufacturer
MIDI	(Musical Instrument Digital Interface) is a serial interface built into electronic music systems
MP3	compressing a sound sequence into a very small file, to enable digital storage and transmission
MP4	a multimedia format designed to store audio-visual (video) data
Normalisation	the process of restructuring data in a database in order to reduce data redundancy and to improve data integrity
OOP	(Object Orientated Programming) a programming language model organised around objects rather than actions and data rather than logic
PaaS	(Platform as a Service) a cloud computing model that delivers applications over the Internet. In a PaaS model, a cloud provider delivers hardware and software tools (usually those needed for software development) to its users as a service
Privacy	setting up the rules so that only those who may see data, do (ensuring security)
Procedural Language	specifies a series of well-structured steps and procedures within its programming context to compose a program. It contains a systematic order of statements, functions and commands to complete a computational task or program
RBD	(Refreshable Braille Display) a peripheral device that allows blind or visually impaired people to interact with a computer. A braille monitor uses the braille system that blind people use to read. Raised dots spell out text that the user
SaaS	(Software as a Service) any cloud service where consumers are able to access software applications over the Internet.
SDRAM	(synchronous DRAM) a generic name for various kinds of dynamic random access memory (DRAM) that are synchronised with the clock speed that the microprocessor is optimised for. This tends to increase the number of instructions that the processor can perform in a given time
Security	Ensuring that data is kept from those who should not see it
SRAM	(static RAM) RAM that retains data bits in its memory as long as power is being supplied. Unlike DRAM (which stores bits in cells consisting of a capacitor and a transistor) SRAM does not have to be periodically refreshed

Solid state	an all-electronic storage device that is an alternative to a hard disk, but faster, because there is zero latency (no moving read/write head)
Sound sampling	receiving analogue sound signals and storing a digital value at a fixed rate
TCP/IP	(Transmission Control Protocol/Internet Protocol) the basic communication language or set of rules of the Internet
Translator	software used to translate a programming language into machine language
Von Neumann model	an architecture that has only one bus which is used for both data transfers and instruction fetches, and therefore data transfers and instruction fetches must be scheduled - they cannot be performed at the same time
WAP	wireless access point
World Wide Web (WWW)	a system of interlinked hypertext documents accessed via the Internet.

The National Institute for Educational Development
P/Bag 2034
Okahandja
NAMIBIA

Telephone: +264 62 509 000
Facsimile: +264 62 509 073

E-mail: info@nied.edu.na
Website: http://www.nied.edu.na