

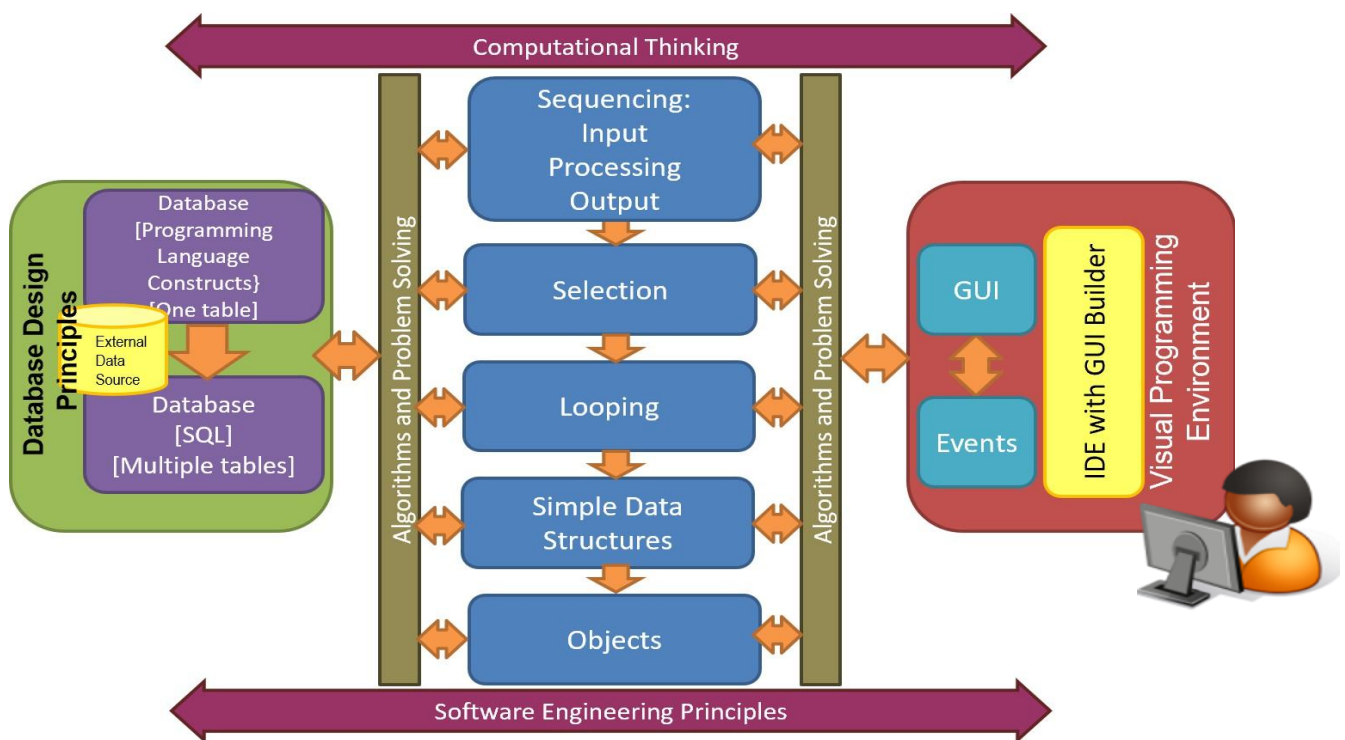
SECTION 3

CONTENT AND SCOPE PER TOPIC

3.1 Solution Development

Solution development is the development of software in a planned and structured process and is based on solving computational problems which include data-related problems through logical thinking. It involves the practices of algorithm development and creating a software solution according to a set of rules and/or requirements specified in the problem statement or by a client/business/individual. The software is developed using appropriate problem-solving techniques, tools and methodologies. Software solution development is achieved through computer programming which could be based on a single or combination of development paradigms such as event-driven programming, object-oriented programming and sequential programming.

Broad topic layout and progression



Note:

In Grade 10 learners are introduced to important computational skills and concepts, algorithm development, problem solving and programming using a high-level programming language that uses an integrated development environment with a GUI builder. Learners are introduced to controls and code and event handling principles.

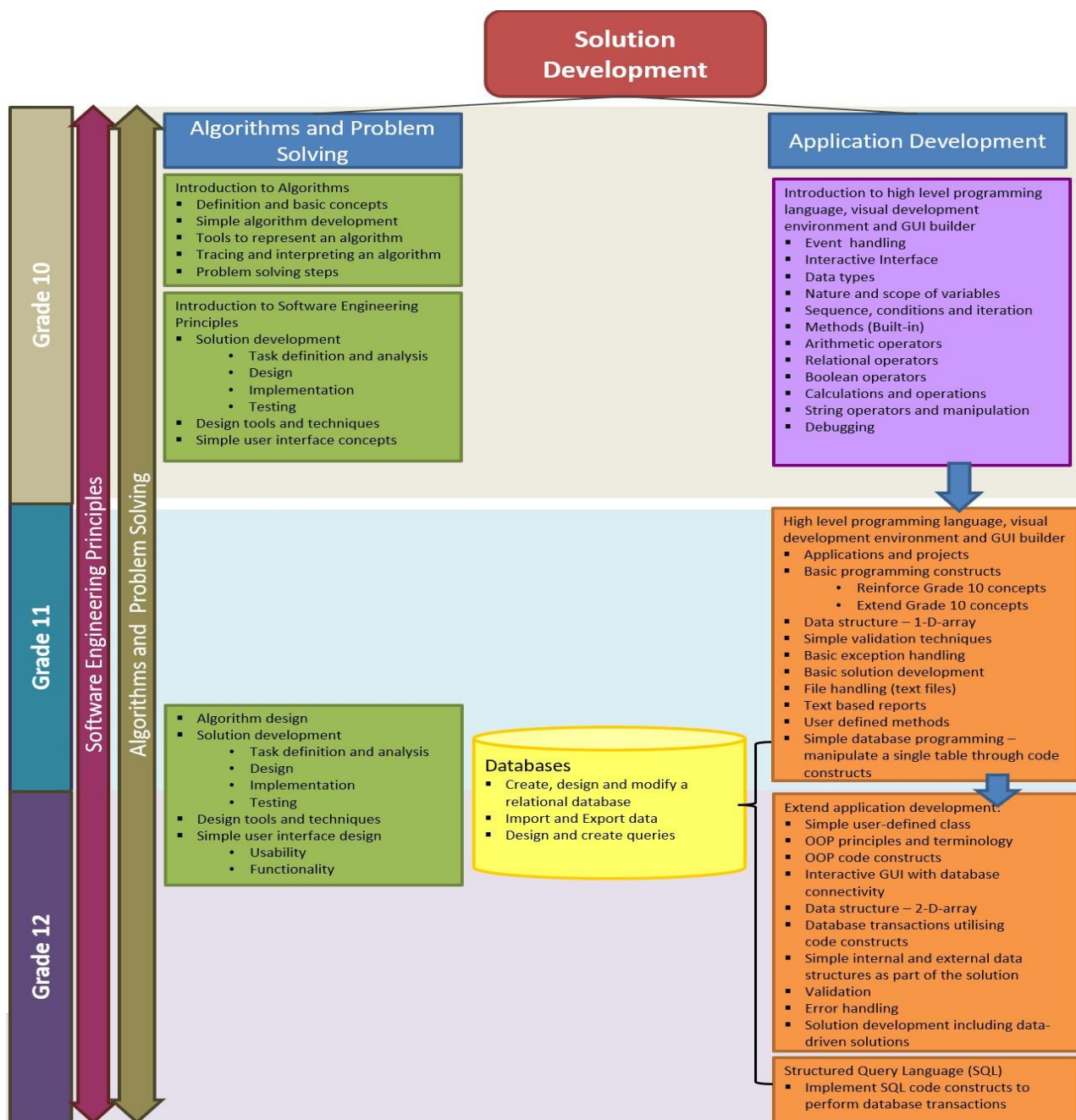
In Grade 11, learners build on the principles and concepts learned in Grade 10. Event handling principles are reinforced using the form class, attributes, methods and controls. Skills to manipulate a database through code constructs are also introduced in Grade 11.

In Grade 12, the principles and constructs are further emphasised through more advanced concepts and problems. Learners are introduced to basic object-oriented programming (OOP) and should be ready to engage with basic structured query language (SQL) code and manipulating a relational database.

The development of computational thinking practices of algorithm development, problem solving and programming underpin solution development and should be emphasised from Grade 10 to Grade 12.

Usability, HCI (human computer interaction) and software engineering principles should be reinforced as part of software development.

Sub-topic layout and progression for Solution Development



Note:

Algorithmic problem solving in Grade 10 should be dealt with separately at first as an introduction to solution development to develop the learner's computational thinking practices of algorithm development, problem solving and programming using everyday scenarios.

Learners should develop an understanding of the importance of order and precision when developing an algorithm as well as the place of algorithms in software solutions and computing science. Thereafter it should be reinforced, extended and integrated with solution development and programming.

Solution development includes computational thinking and the application of software engineering principles using event-driven programming within the object-oriented (OO) paradigm.

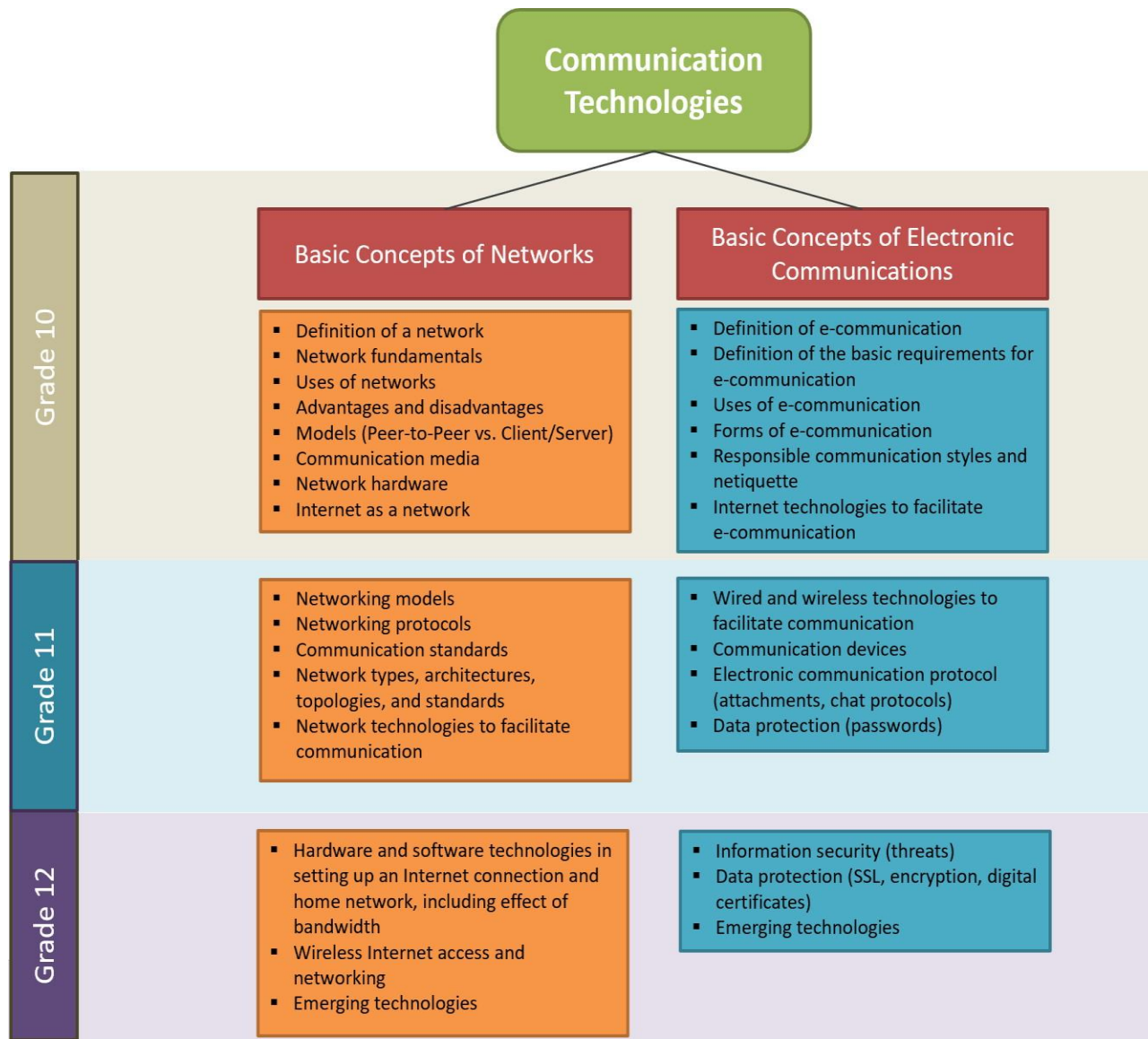
Learners should be able to use appropriate practices and tools to:

- solve computational problems through:
 - identifying and analysing requirements for a specific problem;
 - designing effective algorithms;
 - converting these to code; and
 - testing the solution to see if it meets the requirements.
- apply the principles of human computer interaction to design functional user interfaces.

3.2 Communication Technologies

Communication technologies include various network technologies to facilitate the management and dissemination of digital data from one point to another. Communication technologies also refer to the electronic systems used for electronic data interchange that facilitate, among others, communication and information dissemination between various individuals or groups at a single point or dispersed locations.

Broad topic layout and progression



Note:

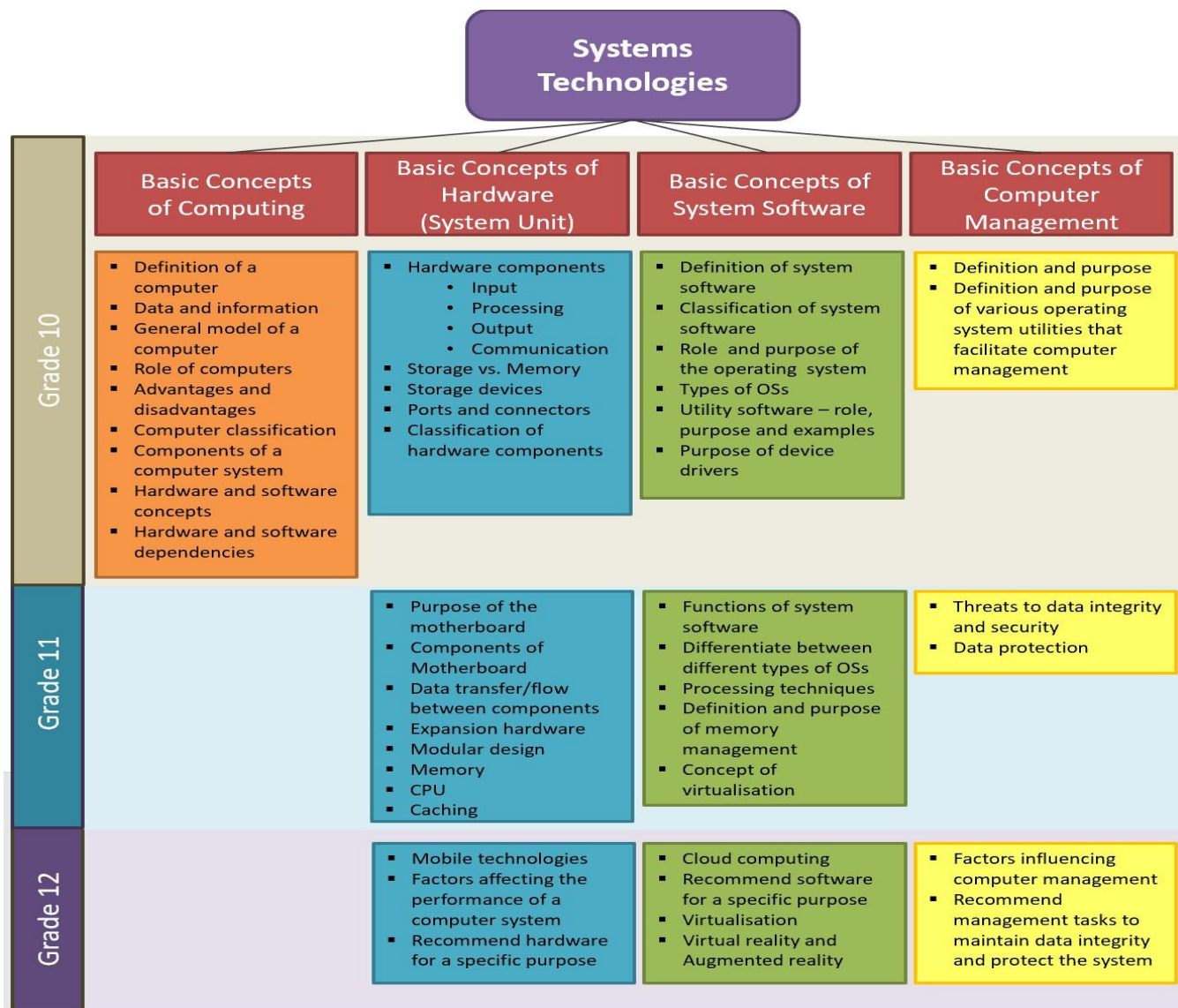
Communication Technologies should teach learners to:

- understand the concepts of the various technologies, standards and protocols involved in the electronic transmission of data via a computer-based network;
- understand the concepts of the technologies and standards implemented to enable electronic communication;
- understand the purpose and uses of communication software;
- understand how communications technology can benefit specific scenarios;
- be aware of and manage safety and security issues; and
- be aware of new trends, developments and technologies.

3.3 Systems Technologies

Systems technologies refer to the physical and non-physical components of a computer system. The components of the system are generally related but unconnected in their original form. The connected components which include hardware, peripherals and software components allow the computer to perform the basic functions of a computing system. The basic functions of a computing system include input, processing, output, storage, communication and transfer of data in an electronic format.

Broad topic layout and progression



Note:

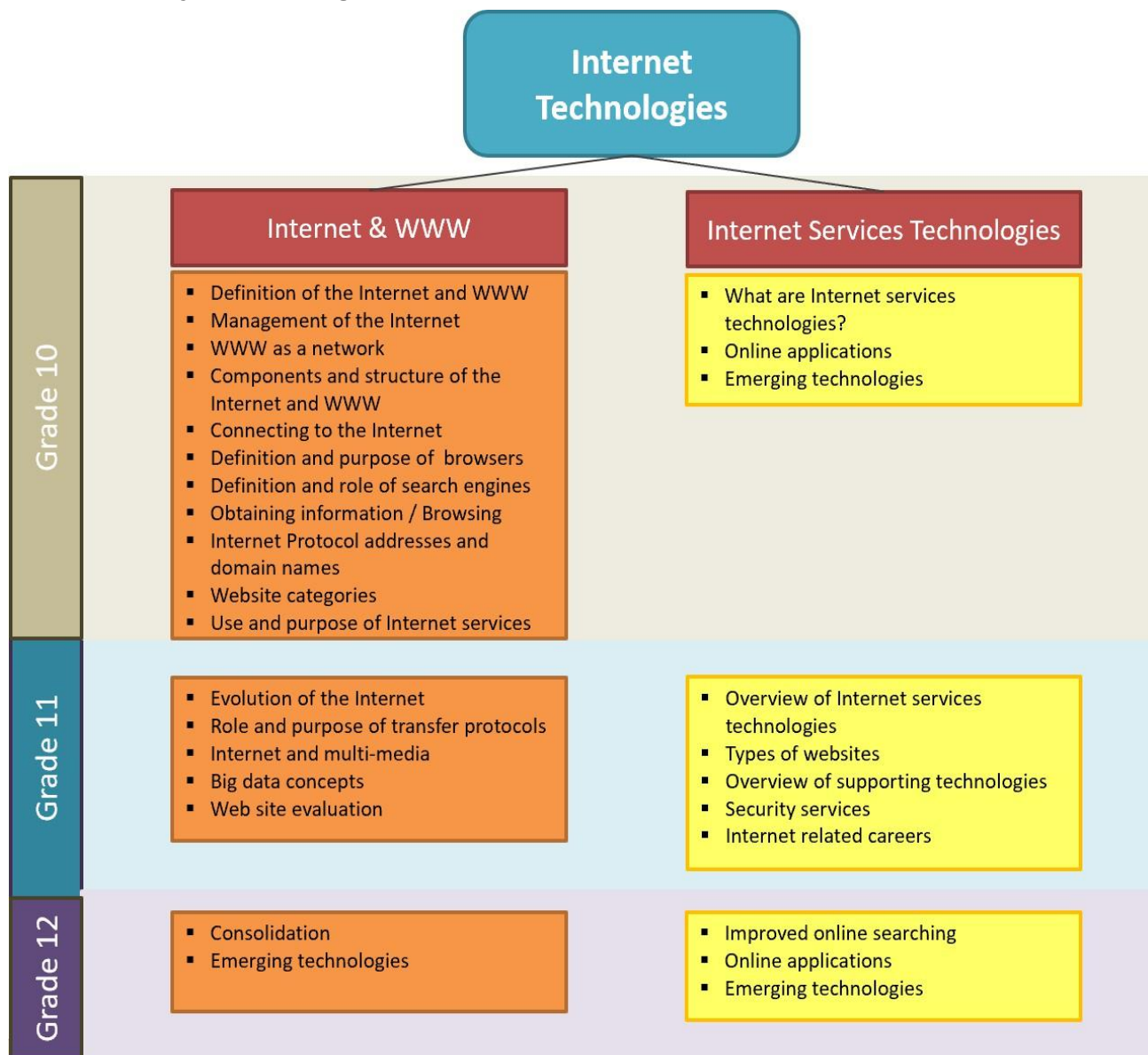
Systems Technologies should teach learners to:

- understand the hardware and software concepts that make up a computer system;
- make informed purchase decisions and whether to upgrade or buy new equipment or software;
- select the most appropriate hardware and software for a given scenario;
- understand how technology can benefit the user in specific contexts;
- understand the operations involved in the management and optimal utilisation of a computer system;
- troubleshoot at an elementary level; and
- be aware of new trends, developments and technologies and how to integrate these with existing or new equipment.

3.4 Internet Technologies

Internet Technologies are related and interconnected technologies which enable the establishment of global networks, for various purposes such as collaboration, electronic data interchange, electronic commerce and social networking. Internet services technologies refer to a range of technologies and tools for the design, development and maintenance of websites. The field of Internet services technologies includes Internet programming as well as the roles and responsibilities of each of the individuals involved. Internet technologies include the WWW and all interrelated processes in the digital presentation of multimedia data on a web page.

Broad topic layout and progression



Note:

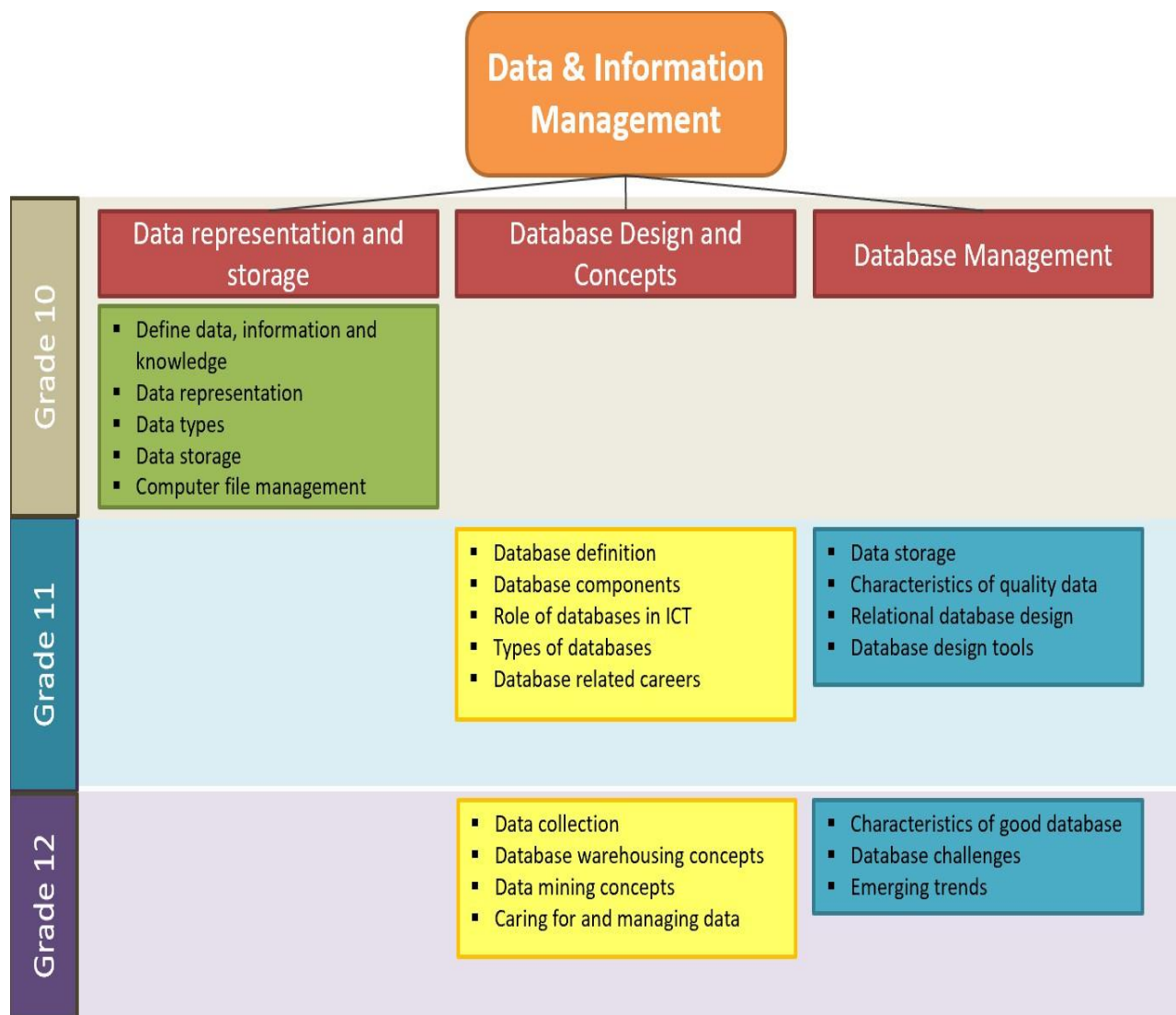
Internet Technologies should teach learners to:

- understand the role that the Internet and the WWW play as part of the global information super-highway and the contribution towards the digital age;
- understand the role of Internet services and supporting technologies;
- understand how Internet technology and services can benefit specific scenarios; and
- be aware of new trends, developments and technologies.

3.5 Data and Information Management

Data and information management refers to the techniques and technologies involved in the collection, storage, dissemination and processing of data into information that results in knowledge and leads to decision making. It includes database design principles with specific reference to data storage, retrieval and information presentation design.

Broad topic layout and progression



Note:

Learners need to develop an understanding of:

- data and information with regard to the representation and classification thereof;
- how business takes advantage of computer databases to store data and retrieve information that enables it to gain a competitive edge as well as the social, legal and ethical issues involved;
- database design for use as part of information-driven ICT systems and platforms; and
- DBMS software and its purpose and application in an information-driven society.

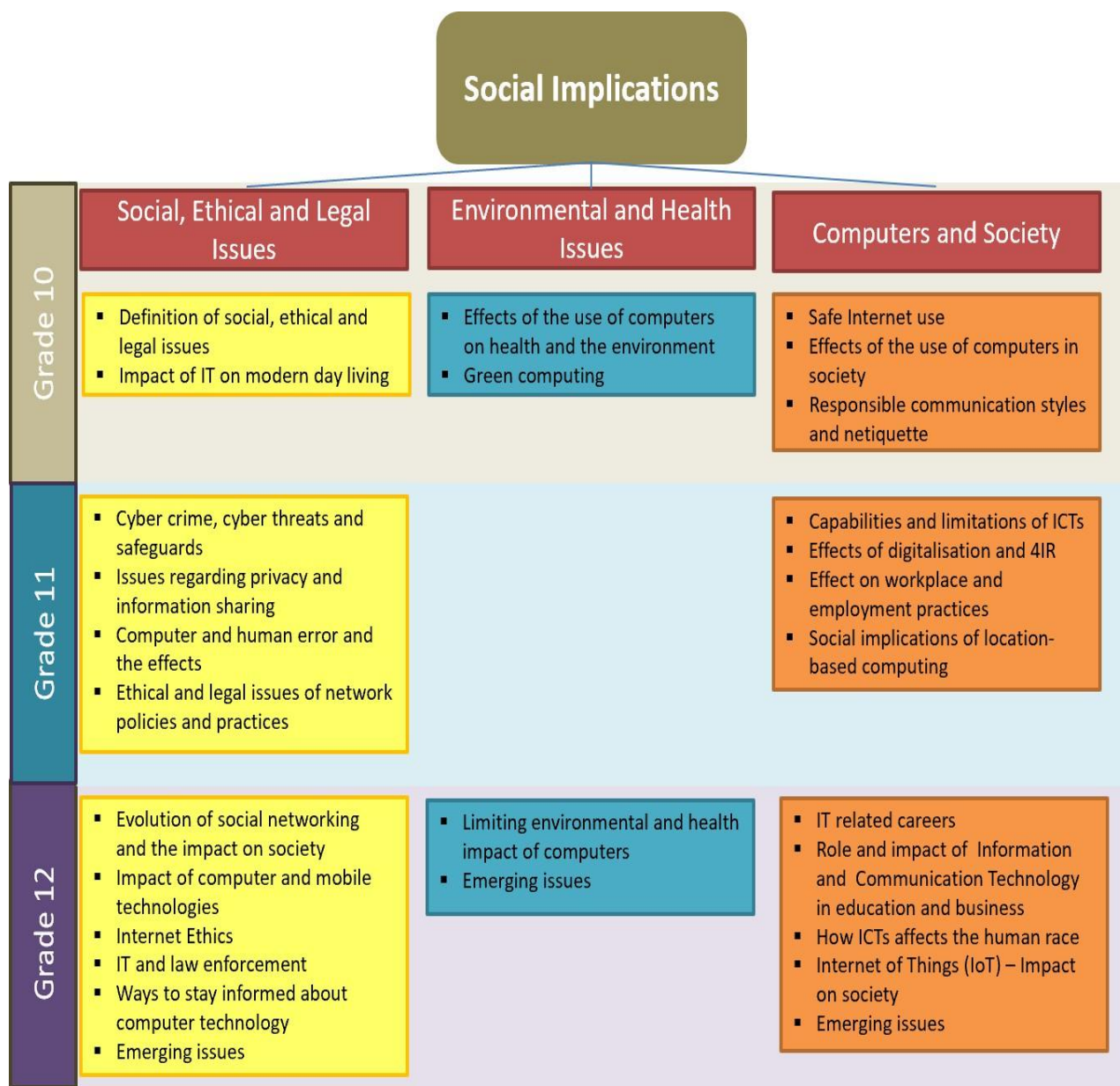
Database design, queries and reports should be linked to application development as described in the topic Solution Development.

This section also covers some practical aspects regarding learning about and working with databases.

3.6 Social Implications

Social implications in the IT curriculum refer to issues relating to the digital age, bridging the digital divide and the responsible use of ICTs.

Broad topic layout and progression



Note:

This topic should provide an overview and understanding of:

- social issues related to the use of computers and how ICTs affect modern life;
- risks and safety aspects that may be involved in the operation of computing equipment within a given context;
- risks and safety issues relevant to using the Internet; and
- principles for making informed decisions regarding the responsible use of ICTs.

Most of the content of Social Implications should be dealt with and integrated with other topics and should not be taught as a stand-alone topic. The time scheduled for this topic can therefore be added to other topics.

3.7 Suggested teaching plan

The suggested teaching plan indicates the minimum content to be covered per term. The sequence of the content or topics listed per term **is not prescribed**. Teachers should design their own work schedules (or use/adapt the work schedule provided in their textbook) to teach the content per term in **appropriate sequence** and pace.

The sub-topics presented in the term plans should not be seen as stand-alone topics. Relevant sub-topics or content should be presented in an integrated manner. Integrating the topics in the lesson presentation should flow naturally due to the nature, links and 'overlap' of the content. Some content from one sub-topic may strengthen and underpin the content of another. This approach should be applied throughout the three-year curriculum.

It is important that the specific technologies in the teaching plans are revised at regular intervals to phase out old technologies and to include new technologies.

As the length of terms varies from one year to another, the teaching plan/work schedules should be adapted accordingly on a year-to-year basis.

3.7.1 Grade 10

Grade 10: Term 1 – 10 weeks/40 hours

Systems Technologies: Basic concepts of computing (± 1 week/4 hours)

- What are Digital Technologies?
- Define Information Technology
- Overview of a general model of a computer in terms of input, storage, processing, output and communication
- Overview and concepts of the main components of a computer system:
 - Hardware
 - Input (keyboard, mouse), storage and memory (hard drive and memory), processing (CPU), output (monitor, printer) and communication (NIC)
 - Software
 - System software (operating system) and application software
 - Generic/common examples and uses
 - What is shareware, freeware, free open-source software (FOSS), proprietary software?
 - Concept of interdependency of hardware and software
 - Hardware vs Software
- Overview of Types of computers (purpose and uses): desktop, laptop, tablets, smartphones, server, embedded computers (microcontrollers)
 - Differentiate between the types of computers in terms of primary uses, processing power and size
 - Categorise computers/classification of computers in terms of portability/mobility, processing power and usage
- Overview of Data and information:
 - Explanation of and differentiation between data and information
 - Information processing cycle: input, processing, output, storage, communication (general concepts)
 - Overview of uses and examples of information within an organisation
 - Why is information useful?
- What is an ICT system?
 - Overview of a general model of an ICT system: convey, manipulate and store data
 - Example of an ICT system (point-of-sales system, mobile phones)

Data and Information Management: Data representation and storage (±2 weeks/8 hours)

- Overview and link between data, information and knowledge
 - What is data representation?
 - What is data storage?
 - Bits and bytes
- Overview of number systems: decimal, binary, hexadecimal
- Conversion between number systems
 - binary and decimal and vice versa
 - decimal and hexadecimal and vice versa
- Overview of digital character representation; ASCII & Unicode
- Overview of data types and their storage
 - Primitive (integer types, character types, floating point types, Boolean)
 - Other data types (text/string types)
- Overview of data structures and collections of data storage in terms of:
 - Files, databases
 - Reasons for data storage
 - Computer file management:
 - Organising files
 - Files, folders and drives
 - File specification: drive, path, filename, file extension
 - File manager

<ul style="list-style-type: none"> ▪ Hierarchical structure ▪ Reasons for having a file structure ▪ Manipulating files and folders (Cut, copy, paste, move and rename) 	
Social Implications (±½ week/2 hours)	
<ul style="list-style-type: none"> • Software licence agreements (including creative commons), piracy, copyright, copyleft • What are the social, ethical and legal issues pertaining to ICTs? <ul style="list-style-type: none"> ◦ Economic reasons for using computers: saving paper, labour, communication costs, efficiency, accuracy and reliability • Digital divide <ul style="list-style-type: none"> ◦ What is the digital divide? ◦ Reasons for the digital divide ◦ What are digitally enabled citizens? • Digital citizenship (What it is) <ul style="list-style-type: none"> ◦ Digital footprint • Advantages and disadvantages of using computers 	
Solution Development: Introduction to Algorithms (±2 weeks/8 hours)	
<ul style="list-style-type: none"> • Basic concepts of an algorithm <ul style="list-style-type: none"> ◦ What is an algorithm? Develop a clear understanding ◦ Examples of algorithms in everyday life, e.g. instructions to draw a kite or fold a paper jet, recipe to bake a cake ◦ Devise an algorithm / basic instruction to complete similar everyday tasks • Basic IPO table & flow charts <ul style="list-style-type: none"> ◦ Tools: basic IPO table & flow charts to represent an algorithm ◦ Interpret a basic IPO table & flow chart ◦ Develop a basic IPO table & flow chart • Examples of algorithms that need to be developed: <ul style="list-style-type: none"> ◦ Determine smallest, largest value of more than two values ◦ Swapping values ◦ Determining aggregates: sum, average (without loops) ◦ Basic calculations: calculating area, volume, VAT ◦ Determine whether a number is even ◦ Determine whether a number is a factor of another number • Produce an algorithm to solve a problem • Trace an algorithm to determine the outcome– trace table <ul style="list-style-type: none"> ◦ Value of accurate, well-tested algorithms <p>Compare algorithms in terms of sequence, precision and efficiency</p>	<p>Notes</p> <p>The purpose of this section is to serve as an introduction to solution development to develop the learner's computational thinking practices of algorithm development, problem solving and programming using everyday scenarios.</p> <p>Exploring algorithms to solve generic problems will enable a learner to use similar principles to devise algorithms for new problems or situations. It will also enable the learner to identify the types of problems requiring certain generic algorithms.</p> <p>Investigating specific algorithms should provide the learner with the opportunity to explore various ways to solve the same problem by using different principles or tools.</p>
Solution development: Introduction to solution development (±2 ½ weeks/ 10 hours)	
<ul style="list-style-type: none"> • Introduction to the programming, basic terms and development environment • Introduction to <ul style="list-style-type: none"> ◦ Output ◦ Input <ul style="list-style-type: none"> ▪ Keyboard ◦ Variables <ul style="list-style-type: none"> ▪ Variable naming conventions ▪ Assigning values to variables ▪ Exploring data types: integers, strings, floats, Boolean ◦ Operators (+, -, *, /) and order of precedence ◦ Retrieving remainders: modulus ◦ Comparison operators and performing logical comparisons ◦ Functions - random, round, square root 	<p>Applying algorithms such as swapping values, finding aggregates, isolate digits in an integer number, finding the smallest/biggest of two numbers, determine if a number is a factor of another number, determine if a number is even</p>

<ul style="list-style-type: none"> ○ Basic calculations such as area, volume, VAT and simple formulae, typical calculations done in other subjects ○ Basic conditional constructs (if and if-then-else) 	
Solution Development: Introduction to solution development using a high-level programming language (±2½ weeks/ 8 hours)	
<ul style="list-style-type: none"> • Introduction to the programming tool – IDE/GUI, basic terms and development environment • Introduction to components (input/ output) • Casting • Formatting of output (fixed, currency) • Event handling (click) 	Notes Delphi: Casting: (StrToInt, IntToStr, FloatToStr, StrToFloat, StrToFloatF)
Formal Assessment (PoA): <ul style="list-style-type: none"> • 1 Theory test 	Refer to Section 4 for: <ul style="list-style-type: none"> • Mark and time allocation • SBA weighting • Term reporting

Grade 10: Term 2 – 10 weeks/40 hours, including examination (2 weeks)
Systems Technologies: Basic concepts of hardware (± 1 week/4 hours)
Describe hardware <ul style="list-style-type: none"> • Extend hardware concepts <ul style="list-style-type: none"> ○ Input devices: (What is it? Where is it used? What is it used for?) <ul style="list-style-type: none"> ▪ Keyboards, pointing devices, scanners, pen, microphones, biometric devices, sensors ○ Output devices: (What is it? Where is it used? What is it used for?) <ul style="list-style-type: none"> ▪ Monitors ▪ Printers (Ink, laser, 3-D) ▪ Compare printers according to features, type of product, quality of output and speed where applicable ▪ Headphones/speakers ○ Storage devices: (What is it? Where is it used? What is it used for?) <ul style="list-style-type: none"> ▪ Hard drives, USB flash drives, solid state drives (SSD), SD cards <ul style="list-style-type: none"> – Compare storage devices in terms of capacity, portability, technology used (magnetic, electronic) ○ Input/Output devices: (What is it? Where is it used? What is it used for?) <ul style="list-style-type: none"> ▪ Touch screens, game controllers, smartphone ○ System unit <ul style="list-style-type: none"> ▪ General function of Motherboard, CPU and RAM ▪ Location of CPU and RAM ○ Ports and connectors: (identify and purpose) <ul style="list-style-type: none"> ▪ USB, HDMI, NIC ○ Categorise hardware according to input, output, storage, processing • Differentiate Primary Memory vs Secondary Memory (Storage) • Compare input, processing, output, storage devices of a desktop computer with a small mobile device, e.g. smartphone/ tablet <ul style="list-style-type: none"> ○ Which are the same? Which are different? Why are they the same/different?
Systems Technologies: Basic concepts of system software (± 1 week/4 hours)
<ul style="list-style-type: none"> • Describe system software • Extend system software concepts <ul style="list-style-type: none"> ○ Operating system <ul style="list-style-type: none"> ▪ What is an operating system? ▪ What is the purpose/role of an operating system? <ul style="list-style-type: none"> – General role: group of related programs which manage hardware and software

<ul style="list-style-type: none"> – Specific role: provides user interface, I/O management – Brief overview of the role of the operating system in terms of file, disk, memory, storage and process management ▪ Types of operating systems (also associate with types of computers), e.g. stand-alone (home edition), network, embedded ▪ Examples of common operating systems (Windows, Linux, iOS, Android) • Utility programs <ul style="list-style-type: none"> o What are utility programs? <ul style="list-style-type: none"> ▪ What are they used for? o Device drivers <ul style="list-style-type: none"> ▪ Definition of a device driver ▪ Installation: manual vs plug-a-play
Communication Technologies: Networks (Overview) (±½ week/2 hours)
<ul style="list-style-type: none"> • Describe a network • Reasons for using networks: <ul style="list-style-type: none"> o Communication, access to/sharing resources, centralisation, file and funds transfer, productivity, leisure • Advantages and disadvantages of networks • List the essential basic network components: <ul style="list-style-type: none"> o Nodes (printer, servers, workstations) o Network Interface Card (NIC) (wired and wireless) o Communication media (wired and wireless) o Switch and Router o Network Operating System (NOS) • Overview of different communication media (wired/wireless) <ul style="list-style-type: none"> o Transmission medium (UTP, fibre optic, radio waves, infrared) o Compare communication medium in terms of accessibility, coverage and security • Personal area network (PAN), Home area network (HAN), Local area network (LAN/WLAN) and wide area network (WAN) (coverage and where it is used) • Internet as an example of a network (WAN) • Differentiate between client-server and peer-to-peer networks <ul style="list-style-type: none"> o Access control
Communication Technologies: Electronic Communications (±½ week/2 hours)
<ul style="list-style-type: none"> • Describe electronic communication • Overview of applications/tools to facilitate e-communication – purpose and uses (What is it? What is it used for?) <ul style="list-style-type: none"> o E-mail, web browser, Cloud storage, instant messaging, chat rooms, video call and Voice over Internet Protocol (VoIP), Vlog, Blog, webinars • E-mail as a form of e-communication <ul style="list-style-type: none"> o Uses of e-mail o E-mail accounts (web-based) <ul style="list-style-type: none"> ▪ Create and Access o E-mail addresses
Social Implications (±½ week/2 hours)
<ul style="list-style-type: none"> • Basic understanding of the following concepts <ul style="list-style-type: none"> o Ergonomics o Green computing issues o Health issues • E-communication in terms of accuracy, time, distance, communication costs, speed • Responsible communication styles and netiquette • How to use e-mail (best practices) <ul style="list-style-type: none"> o Use of To, Cc, Bcc, forward, reply, attachments and adding a subject to an e-mail

Solution Development: Software Engineering Principles (±1½ week/6 hours)	
<ul style="list-style-type: none"> • What is problem solving? • Problem solving steps (<i>Polya, G., 1957</i>) <ul style="list-style-type: none"> ○ Understand the problem (task/problem description) <ul style="list-style-type: none"> ▪ State in own words ▪ Clarity on what needs to be done ▪ What is known or given? What is missing or needed? ○ Devise an algorithm look for patterns <ul style="list-style-type: none"> ▪ Look at related problems, known solutions ▪ Examine simpler or special cases ▪ Make a table, create diagram, use guess and check, work backwards, identify sub-goal (identify smaller tasks) ▪ Present algorithm using an appropriate tool (flowchart or pseudo code) ○ Carry out the plan/implement the algorithm (write the code) ○ Look back/test (see if it works – trace tables) <ul style="list-style-type: none"> ▪ Check results against original problem. Does it make sense? Is there another solution? • Apply problem solving techniques (as above) <ul style="list-style-type: none"> ○ Examples: <ul style="list-style-type: none"> ▪ Determine smallest, largest value of more than two values ▪ Determine whether a number is even ▪ Determine whether a number is a factor of another number ▪ Solve any other relevant problems • Use appropriate tools and techniques used in software analysis, viz.: <ul style="list-style-type: none"> ○ User stories (written by the client and provide the requirements) ○ Noun-verb analysis of user stories ○ List of nouns provides identification of objects and state ○ List of verbs provides identification of behaviour ○ Acceptance tests (does the program meet the requirements?) • Implement algorithms to solve general computing problems using code constructs covered during term 1 and 2. 	
Solution Development: Introduction to solution development using a high-level programming language (±4 weeks/16 hours)	
Apply the following using code constructs <ul style="list-style-type: none"> • Comparison operators and performing logical comparisons (from term 1) • Conditional constructs (if and if-then-else) including Boolean operators <ul style="list-style-type: none"> ○ Nested if's ○ CASE statement • Extend the use of variables, relational operators • Boolean logic/operators (AND, OR, NOT, IN) • String comparisons (Basics) • Basic validation techniques (input and processing), e.g. test for negative number when calculating square root • Events – form create activate • Debugging techniques • Debugging using trace tables 	Delphi: String methods – length, UpperCase, UpCase, Trim (to determine the length, comparing length, change to uppercase, comparing strings)
Assessment (PoA): <ul style="list-style-type: none"> • A Practical test to be administered during the term • Mid-year examination (1 practical paper + 1 theory paper). 	Refer to Section 4 for: <ul style="list-style-type: none"> • Mark and time allocation • SBA weighting • Term reporting

Grade 10: Term 3 – 10 weeks/40 hours
Systems Technologies: Computer Management (±1 week/4 hours)
<ul style="list-style-type: none"> • Describe computer management • Overview and purpose of various management tasks and operating system utilities <ul style="list-style-type: none"> ○ Management of desktop ○ Management of files and folders <ul style="list-style-type: none"> ▪ File-naming conventions ▪ Common file types and extensions (association) ▪ Compressed ▪ Forms of text files ▪ Database, spreadsheet, presentations and word processing documents ▪ Graphic files, movie, sound and animation files ▪ Source code (Delphi only) ▪ Object code, executable files and shared and dynamically linked libraries ▪ Saving as another type/version and exporting between file types ○ General housekeeping tasks <ul style="list-style-type: none"> ▪ Disk Clean-up ▪ Updating ▪ Archive, backup ▪ Compress/decompress files ▪ Security features – firewall, anti-virus, control of spyware, adware ▪ Installing/uninstalling software (custom and full installation, License keys, activation codes) ▪ System settings (regional and language)
Internet Technologies: Internet and WWW (±1½ week /± 6 hours)
<ul style="list-style-type: none"> • Overview of the Internet <ul style="list-style-type: none"> ○ Describe the Internet ○ Internet addresses – Internet protocol (IP) addresses and domain names • What is needed to connect to the Internet referring to <ul style="list-style-type: none"> ○ Internet Service Providers (ISPs), wired and wireless connections • Overview of the World Wide Web (WWW) <ul style="list-style-type: none"> ○ Describe the WWW ○ Web address/uniform resource locator (URL) ○ Web page and website • Browsing and searching <ul style="list-style-type: none"> ○ What is a web browser ○ Examples of web browsers ○ How to access and browse a website ○ What is a search engine? ○ Examples of search engines ○ Performing searches using a search engine (search techniques)
Social Implications (±½ week/2 hours)
<ul style="list-style-type: none"> • E-mail & Internet threats and issues: viruses, hoaxes, fake news, spam, phishing, e-mail spoofing and pharming • Safe email and Internet use: dangers and tips to ensure safe use • POPIA (Basic understanding and purpose)
Solution Development: Introduction to solution development using high level programming language (±5 weeks/ 20 hours)
<p>Using good programming principles and algorithms - extend the use of planning tools and techniques:</p> <ul style="list-style-type: none"> • Iteration constructs: fixed, pre-conditional and post-conditional (for-loop, while and repeat-until) • String handling from first principles (no built-in methods)

<ul style="list-style-type: none"> ○ Count the number of occurrences of a specific character in a string ○ Determine the position of a character in a string ○ Remove a specific character from a string ○ Replace a specific character in a string ○ Add a character in a specific place in a string ○ Extract a character or a subset of characters from a string ● Apply String methods to String handling <ul style="list-style-type: none"> ○ length, left, right, position, copy, trim, delete, insert ● Implement algorithms to solve the following computing problems: <ul style="list-style-type: none"> ○ Determine whether a number is a prime number ○ Lowest common multiple (LCM), greatest common divisor (GCD) ○ Find a specified character in a string ● Develop simple applications incorporating a combination of graphics, iteration, conditional constructs, concepts covered ● Make use of a timer object for simple animations 	
Solution Development: Software Engineering Principles and Practical Assessment Task (PAT) (±2 weeks/8 hours)	
Program persistence (to retrieve the previous state of a program) Project requirements <ul style="list-style-type: none"> ● The problem statement ● Research ● User stories: Who, What, Limitations ● Diagram that gives overview of the program functionality GUI design <ul style="list-style-type: none"> ● Apply HCI principles: Grouping of components ● Include panels and/or tabbed panes (no multiple screens) Implement each requirement <ul style="list-style-type: none"> ● IPO ● Design flowchart ● List possible test cases link to user stories ● Develop the Code to implement the requirements ● Test the code 	Develop simple applications incorporating a combination of graphics, iteration, conditional constructs, concepts covered.
Assessment (PoA): <ul style="list-style-type: none"> ● 1 Alternative task: Closed or open book, OR case study OR integrated task. ● 1 Practical test 	Refer to Section 4 for: <ul style="list-style-type: none"> ● Mark and time allocation ● SBA weighting ● Term reporting

Grade 10: Term 4 – 10 weeks/40 hours, including examination (3 weeks)	
Internet Technologies: Internet and WWW (±½ week/2 hours)	
<ul style="list-style-type: none"> Internet services technologies (What is it? What is it used for?) <ul style="list-style-type: none"> Online: Storage, Synchronisation, Collaboration Online Shopping Social media 	
Solution Development: Introduction to solution development (±1 weeks/ 4 hours)	
<p>Using good programming principles and algorithms - extend the use of planning tools and techniques:</p> <ul style="list-style-type: none"> Revise, consolidate and extend solution development content by developing applications incorporating a combination of structures and features 	<p>Solve various computational problems through identifying and analysing requirements for a specific problem; designing effective algorithms; converting these to code and testing the solution to see if it meets the requirements.</p> <p>In doing so, learners should make use of appropriate practices, tools and techniques and consider HCI principles in designing user interfaces.</p>
Solution Development: Software Engineering Principles and Practical Assessment Task (PAT) (±3 weeks/12 hours)	
<p>Input and output using a text file (PAT only - optional)</p> <ul style="list-style-type: none"> Apply simple file input and output using a text file Read from a file to retrieve data and populate a list box Save data to a file from a list box <p>Finalise PAT</p> <ul style="list-style-type: none"> Construct a solution based on the planning <ul style="list-style-type: none"> Document the solution by adding comments 	<p>Delphi:</p> <p>Use only LoadFromFile and SaveToFile to retrieve and save data/ information</p>
<p>Assessment (PoA):</p> <ul style="list-style-type: none"> Examination <ul style="list-style-type: none"> 1 Practical paper 1 Theory paper <p>Practical Assessment Task (PAT):</p> <p>To be completed before the commencement of end of year examinations.</p> <p>School Based Assessment (SBA)</p>	<p>Refer to Section 4 for:</p> <ul style="list-style-type: none"> Mark and time allocation SBA weighting <p>Reporting (promotion mark)</p> <p>Final Practical paper – 20%</p> <p>Final Theory paper – 20%</p> <p>PAT – 20%</p> <p>SBA – 40%</p>

Grade 11: Term 1 – 10 weeks/40 hours
Systems Technologies: Hardware (±1 week/4 hours)
<p>Extend hardware concepts from Grade 10:</p> <ul style="list-style-type: none"> • Describe the motherboard • Purpose and role of the motherboard • Components as part of the motherboard <ul style="list-style-type: none"> ◦ Purpose and role of a BIOS chip, CPU, GPU, RAM, VRAM, ROM, slots and busses ◦ Modular design • Purpose and role of the expansion cards • Flow/transfer of data between components <ul style="list-style-type: none"> ◦ Busses (USB) ◦ Point-to-point connections ◦ Storage → RAM → CPU ◦ RAM → VRAM → GPU • Purpose and role of cache memory and caching <ul style="list-style-type: none"> ◦ Memory as part of a computer system ◦ ROM, RAM – role and characteristics • Computer performance <ul style="list-style-type: none"> ◦ Difference in performance of different components and caching (cache memory, web caching and disk caching)
Systems Technologies: Software (±1 week/4 hours)
<p>Extend functions of system software from Grade 10:</p> <ul style="list-style-type: none"> • Various types of operating systems in terms of cost, size, hardware needed and platform • What are programming language compilers/interpreters? • Overview of processing techniques (managed by systems software) <ul style="list-style-type: none"> ◦ Multi-tasking, multi-threading, multi-processing (definition, comparison) • Virtual memory (Role and purpose) • Overview of virtualisation <ul style="list-style-type: none"> ◦ Describe virtualisation ◦ Virtual machines – purpose
Communication Technologies: Networks (±1 week/4 hours)
<ul style="list-style-type: none"> • Overview of physical aspects of a network <ul style="list-style-type: none"> ◦ Data transmission <ul style="list-style-type: none"> ▪ Media (reinforce from Grade 10 and extend (UTP, Fibre, Wi-Fi, 4G/5G, Satellite)) ▪ Physical layout (topology – star) ▪ Physical limitations (access and bandwidth) ▪ Connection devices (NIC, modem, switch, router/bridge) ▪ Compare PAN, HAN, LAN, WAN in terms of composition, media and capacity • Overview of network innovation (role and purpose) <ul style="list-style-type: none"> ◦ Voice over Internet Protocol (VoIP) ◦ Internet vs Intranet vs Extranet ◦ Virtual Private Networks (VPN) ◦ Location-based computing <ul style="list-style-type: none"> ▪ What it is ▪ Applications making use of current location such as weather app, traffic app
Social implications (±½ week/2 hours)
<p>Social issues applicable to term 1 content:</p> <ul style="list-style-type: none"> • The social implications of location-based computing • Ethical and legal issues of network use policies and practices (e.g. AUP) • Capabilities and limitations of hardware and software

Systems Technologies: Computer Management (±½ week/2 hours)	
<p>Extend computer management issues regarding safeguarding against threats</p> <ul style="list-style-type: none"> • Safety and security <ul style="list-style-type: none"> ◦ Human error (GIGO, accidents) • Threats <ul style="list-style-type: none"> ◦ Physical access <ul style="list-style-type: none"> ▪ Theft ▪ Flash drives and portable media ◦ Hardware failure <ul style="list-style-type: none"> ▪ Storage ▪ Power ◦ Network vulnerability (Overview) <ul style="list-style-type: none"> ▪ Malware: Viruses, Worms, Trojans, Ransomware, Spyware, Rootkits, Keyloggers, Bots ▪ Social engineering: spoofing, phishing, pharming • Remedies (Overview) <ul style="list-style-type: none"> ◦ Awareness training ◦ Access controlled rooms, Alarms ◦ Backup (including on-line storage), UPS ◦ Username & Strong passwords, access rights, firewalls, anti-virus, validation (Captcha) 	
Solution Development: Application Development using a high-level programming language (±6 weeks/24 hrs)	
<p>Apply the following using problem solving steps, tools, techniques (from grade 10) and code constructs</p> <p>Nested loops</p> <ul style="list-style-type: none"> • Structure • Algorithms: <ul style="list-style-type: none"> ◦ Exploring algorithms such as converting binary numbers to decimal numbers and vice versa ◦ Use special characters such as a star to draw with code (simple geometrical shapes) <p>Arrays</p> <ul style="list-style-type: none"> • Explain the concept of a 1-dimensional array • Arrays as a data structure (1-D) • Loop through items • Basic mathematical operations, e.g. aggregates (sum, average, product), minimum, maximum • Algorithms: <ul style="list-style-type: none"> ◦ Search using the linear and binary search algorithm ◦ Sorting an array (bubble and selection sort) • Parallel arrays <p>String and Date manipulation</p> <ul style="list-style-type: none"> • String manipulation using string methods: <ul style="list-style-type: none"> ◦ inserting and deleting characters ◦ determine the position of a character ◦ find a character/substring ◦ determine the length of a string • Date and time objects <ul style="list-style-type: none"> ◦ Changing the date and time ◦ Formatting date and time ◦ Date calculations ◦ Date methods: time to string, date to string, test for leap year 	<p>Compare efficiency of listed sorting and searching algorithms e.g. using trace tables.</p> <p>Learners only need to apply ONE of the listed searching and sorting algorithms for practical assessment purposes</p> <p>Built-in methods</p> <p>Extend the use of built-in methods and the concept of parameters/message passing</p> <p>Delphi date methods:</p> <p>TimeToStr, DateToStr, IsLeapYear</p>

Assessment (PoA): <ul style="list-style-type: none"> • 1 Practical test • 1 Theory test 	Refer to Section 4 for: <ul style="list-style-type: none"> • Mark and time allocation • SBA weighting • Term reporting
--	--

Grade 11: Term 2 – 10 weeks/40 hours, including examinations (2 weeks)
Communication Technologies: Electronic Communications (±1 week/4 hours)
<ul style="list-style-type: none"> • Mobile/wireless e-communication (purpose, writing style and audience) <ul style="list-style-type: none"> ◦ E-mail ◦ Microblog, SMS, instant messaging ◦ Social Media: video casting, podcasting, VoIP, video conferencing • Use of mobile technology (What it is) <ul style="list-style-type: none"> ◦ Mobile devices (smartphones, tablets, laptops) ◦ Mobile browser, mobile app – description and feature for example app vs web • Use of wireless technologies <ul style="list-style-type: none"> ◦ GPS, Generation of cellular technology (4G/5G), Bluetooth. ◦ Wi-Fi, Access points ◦ Difference in range and bandwidth (non-technical) ◦ Connection speed <ul style="list-style-type: none"> ▪ Shaping ▪ Throttling • Protocols <ul style="list-style-type: none"> ◦ The function of protocols (POP3, SMTP, IMAP, VoIP) • Data Security <ul style="list-style-type: none"> ◦ Blockchain (Definition and purpose) ◦ Method of practising data security (Passwords, encryption, multi-layer verification/authentication) (Definition and purpose)
Social Implications (±½ week/2 hours)
<ul style="list-style-type: none"> • Effects of digitalisation <ul style="list-style-type: none"> ◦ Gig economy (Definition and examples) ◦ Online services (online banking, booking reservations, e-learning) ◦ Video conferencing, online collaboration, ◦ Social media and influencing ◦ Careers: digital forensic investigator, 3D architects and 3D printing operators (prosthetic limbs), privacy consultant, medical nanobot technician ◦ Effect on workplace and employment practices <ul style="list-style-type: none"> ▪ Remote office, decentralisation of labour, office automation, robotics, artificial intelligence • How to protect your online identity
Data and Information Management: Database Management (±½ week/2 hours)
<ul style="list-style-type: none"> • Describe database management software (DBMS) (Definition and functions) • Examples of DBMS software, <ul style="list-style-type: none"> ◦ Proprietary - Microsoft SQL Server, Oracle, Microsoft Access ◦ Open source – PostgreSQL, MySQL • Database types according to usage requirements – (size and accessibility) <ul style="list-style-type: none"> ◦ Desktop/personal database ◦ Server/centralised database ◦ Distributed database • Overview of database-related careers and roles of people involved (Database Administrator (DBA), Database Programmers, Database Analysts, Database Project Managers)

Solution Development: Application Development using a high-level programming language (±5 weeks/20 hours)	
<p>Using good programming principles and algorithms - extend the use of planning tools and techniques:</p> <p>Introduction to text files</p> <ul style="list-style-type: none"> • What is a text file? • Physical file vs Logical file name • Layout of data in a text file: end of line, end of file characters • Trace tables to demonstrate parameter passing • Text files can be used to populate other data structures • Text file procedures to: <ul style="list-style-type: none"> ◦ Assign a file to a file variable ◦ Overwrite a file ◦ Close a file ◦ Open a file for reading ◦ Read a line of text in file ◦ Write a line of text to the file ◦ Open an existing file to add text • Reading from a text file <ul style="list-style-type: none"> ◦ Exception Handling ◦ Checking whether file exists ◦ Reading one line at a time ◦ Reading multiple lines ◦ Displaying information from a text file ◦ Use of string handling functions in application exercises • Adding to a text file <ul style="list-style-type: none"> ◦ Write to a file using overwrite or append mode ◦ Display the updated text file ◦ Generating a text-based report, e.g. correctly formatted data • User defined methods with and without parameter passing • Introduction to user defined methods <ul style="list-style-type: none"> ◦ Purpose ◦ Types (functions and procedures) ◦ Differentiate between procedures and functions • Procedures and Functions <ul style="list-style-type: none"> ◦ Structure, calling statement and method signature ◦ Arguments vs parameters ◦ Value parameters only ◦ Using methods in problems ◦ Basic input validation techniques ◦ Input validation using code constructs 	<p>Delphi text file procedures: AssignFile, Rewrite, CloseFile, Reset, Readln, Writeln, Append</p>
Solution Development: Software Engineering Principles and PAT (±1 week/4 hours)	
<ul style="list-style-type: none"> • What is software development? <p>Planning and implementing a solution</p> <ul style="list-style-type: none"> ◦ Define/understand the problem/task ◦ Read the specifications and analyse the problem/task to determine the requirements <p>• Design the interface and the solution</p> <ul style="list-style-type: none"> ◦ Develop a logical solution based on the specifications and analysis as well as sound software engineering principles ◦ Consider functionality and usability issues in designing the interface <p>• Code/implement</p>	

<ul style="list-style-type: none"> ○ Incorporate suitable programming constructs in the development of a solution • Test and debug the program <ul style="list-style-type: none"> ○ Use testing and debugging techniques and methods • Document, implement and maintain the program • Planning techniques using any appropriate tools • Dynamic Instantiation of active and passive components (functions and procedures) – GUI design 	
Assessment (PoA): <ul style="list-style-type: none"> • Mid-year examination <ul style="list-style-type: none"> • 1 Practical paper • 1 Theory paper 	Refer to Section 4 for: <ul style="list-style-type: none"> • Mark and time allocation • SBA weighting • Term reporting

Grade 11: Term 3 – 10 weeks/40 hours
Data and Information Management: Database design concepts (±3 weeks/12 hours)
<ul style="list-style-type: none"> • Relationship between data, information, knowledge and decision making • Characteristics of quality data: <ul style="list-style-type: none"> ○ Accuracy, correctness, currency, completeness, relevance ○ Data validation: format check, data type check, range check, check digit • How qualities of valuable information can be used to build knowledge and make decisions • Accessing and manipulating data <ul style="list-style-type: none"> ○ Manual ○ Electronic • Grouping data <ul style="list-style-type: none"> ○ Records and fields ○ Different types of fields and their purpose, e.g. primary key, alternate key, foreign keys, composite keys ○ Tables ○ Relationships (relationships between tables, referential integrity & entity relationship diagrams (ERD)) • Reporting on data • Create a simple database without relationships <ul style="list-style-type: none"> ○ Design the table(s) ○ Maintain data: insert/add/import, delete, edit ○ Process, sort, query (generating information from a database)
Social Implications (±½ week/2 hours)
<ul style="list-style-type: none"> • Discuss the effect of Computer and human error: <ul style="list-style-type: none"> ○ Inaccurate and invalid – data input ○ Unverified data ○ Software bugs ○ Incorrect hardware/software configurations • Discuss the effect of cybercrime: <ul style="list-style-type: none"> ○ Identity theft ○ Business data theft ○ Virus attack to company (ransomware and malware)
Solution Development: Application Development using a high-level programming language (±4½ weeks/18 hours)
<p>Using good programming principles and algorithms - extend the use of planning tools and techniques:</p> <ul style="list-style-type: none"> • Accessing a database through programming language constructs • Set up a connection or connect to a database (single table) by providing path in code statements • Develop a multi-form/multi-screen GUI incorporating simple controls – consider functionality and usability • Use programming language constructs in the execution of various simple database transactions <ul style="list-style-type: none"> ○ Access fields and records within a dataset with code constructs and applicable methods ○ Navigate the records of a dataset

<ul style="list-style-type: none"> ○ Modify individual fields and records within a dataset with code constructs and applicable methods, and apply all changes ○ Manipulate a dataset object and records with code constructs and apply all changes ○ Reinforce concepts such as iteration and conditions • Reinforce methods as part of a solution <ul style="list-style-type: none"> ○ Apply simple parameter passing and return values using class methods as part of the form class • Design and develop solutions for specific problems that include computational thinking and software engineering principles <ul style="list-style-type: none"> ○ Apply generic algorithms as part of the solution ○ Incorporating database transactions managed by methods or events ○ Devise a specific algorithm where applicable to solve a problem utilising user-defined code constructs or built-in methods ○ Motivate the use of a specific algorithm ○ Validate the solution against a set of data using different techniques, e.g. trace tables, watches, manual output comparison 	
Solution Development: Software Engineering Principles and PAT (±2 weeks/8 hours)	
<ul style="list-style-type: none"> • Reinforce problem-solving steps and software engineering principles • Set up relationships between tables (Optional) <ul style="list-style-type: none"> ○ 1:M e.g. register class → pupils ○ Two tables showing master detail relationship with at least one foreign key in one table ○ Primary key and foreign key ○ Simple entity relations diagrams (ERD) • Create a query to extract information from a database using a relationship on a maximum of two tables with multiple criteria 	
Assessment (PoA): <ul style="list-style-type: none"> • 1 Theory test /Alternative Assessment: Closed or Open Book or Case Study or Survey • 1 Practical test 	Refer to Section 4 for: <ul style="list-style-type: none"> • Mark and time allocation • SBA weighting • Term reporting

Grade 11: Term 4 – 10 weeks/40 hours, including examinations (2 weeks)
Internet Technologies: Internet and WWW (± ½ week/2 hours)
<ul style="list-style-type: none"> • Overview of the evolution of the Internet in terms of: <ul style="list-style-type: none"> ○ Software and applications (definition) <ul style="list-style-type: none"> ▪ Internet of Things (IoT) (characteristics and uses) <ul style="list-style-type: none"> – Connected security system – Household appliances – Connected cars ○ 4IR & 5IR overview • Big data concepts <ul style="list-style-type: none"> ○ What is meant by Big data? ○ Characteristics <ul style="list-style-type: none"> ▪ Volume ▪ Variety ▪ Velocity ▪ Value ○ Use <ul style="list-style-type: none"> ▪ Customer profiling ▪ Advance patient care ▪ Agricultural • Using Internet technologies to access multimedia content <ul style="list-style-type: none"> ○ Download vs streaming ○ Live broadcasts • Media

<ul style="list-style-type: none"> ○ Compression technology <ul style="list-style-type: none"> ▪ Lossless (Definition & Application) ▪ Lossy (Definition & Application) ○ Compression: Quality vs bandwidth and speed 	
Internet Technologies: Internet Services Technologies (±1 week/4 hours)	
<ul style="list-style-type: none"> • Overview of Internet services technologies • Types of websites (what they offer) <ul style="list-style-type: none"> ○ Static vs dynamic sites (ability to store data, interactivity, media, advantages and disadvantages) ○ Location based services sites • Overview of supporting technologies: <ul style="list-style-type: none"> ○ The function of HTTP, HTTPS • Security services (purpose, advantages and limitations) <ul style="list-style-type: none"> ○ Multi-factor authentication ○ One time pin (OTP) ○ Security token valid for limited period • Internet related careers (Web designer, Web author, Graphics and multimedia designer) 	
Social Implications (±½ week/2 hours)	
<ul style="list-style-type: none"> • Discuss the social implications of big data • Describe the influences of globalisation and the industrial revolutions (4IR & 5IR) • IoT – Technologies enabling the IoT and their impact on society 	
Solution Development: Software Engineering Principles and PAT (±3 weeks/12 hours)	
<ul style="list-style-type: none"> • Reinforce software engineering principles, algorithms and problem-solving techniques • Practical Assessment Task – finalise 	
Assessment (PoA): <ul style="list-style-type: none"> • Examination <ul style="list-style-type: none"> • 1 Practical paper • 1 Theory paper Practical Assessment Task (PAT): To be completed before the commencement of end of year examinations. School Based Assessment (SBA)	Refer to Section 4 for: <ul style="list-style-type: none"> • Mark and time allocation • SBA weighting Reporting (promotion mark) Final Practical paper – 20% Final Theory paper – 20% PAT – 20% SBA – 40%

Grade 12

Grade 12: Term 1 – 10 weeks/40 hours
Data and Information Management: Database Management (±1 week/4 hours)
<ul style="list-style-type: none">• Data collection – Overview and examples<ul style="list-style-type: none">◦ Forms (registration and application)◦ Tags (RFID, digital sensors)◦ Invisible Online data collection (browsing, mobile phones)◦ Transaction tracking (e.g. credit card, loyalty card)◦ Location based data• Data warehousing<ul style="list-style-type: none">◦ What is data warehousing?◦ Data warehousing vs Database• Data mining<ul style="list-style-type: none">◦ What is data mining?◦ Processes of data mining (extracting data, looking for patterns, discovering knowledge)◦ Practical examples (government social grants, social media)• Caring for and managing data<ul style="list-style-type: none">◦ Value of data and metadata◦ How to protect data: validation, verification, integrity, logging changes – (who, what, when), warehousing, controlling access (passwords, security, user rights), parallel data sets
Data and Information Management: Database design concepts (±2½ weeks/10 hours)
<p>Design and create a relational database (extend from Grade 11)</p> <ul style="list-style-type: none">• Characteristics of a good database<ul style="list-style-type: none">◦ Data integrity◦ Data independence◦ Data redundancy◦ Data security◦ Data maintenance (ease of)• Challenges with databases<ul style="list-style-type: none">◦ Anomalies• How to get rid of anomalies (concept of normalisation)<ul style="list-style-type: none">◦ Split tables and create relations◦ Key fields - reinforce<ul style="list-style-type: none">▪ Primary and alternate keys▪ Foreign keys▪ Composite keys◦ Example of basic relationship enabled by the utilisation of key fields
Systems Technologies: Hardware (±½ week/2 hours)
<ul style="list-style-type: none">• Mobile technologies<ul style="list-style-type: none">◦ Smartphones, laptops, tablets<ul style="list-style-type: none">▪ Advantages of mobility (size, weight)▪ Constraints (battery life, computing power vs. power consumption)• Overview of factors influencing performance of a computer<ul style="list-style-type: none">◦ CPU (speed and multi-processing)◦ GPU (type, speed)◦ Memory capacity (RAM and cache memory)◦ Storage (type, speed and disk caching)◦ Network interface card (speed)

<ul style="list-style-type: none"> • Motivate a typical computer system in respect of user requirements 	
Social Implications (±½ week/2 hours)	
Social issues applicable to term 1 content: <ul style="list-style-type: none"> • Discuss the advantages and disadvantages how mobile technology impacts on privacy – Personal and Business • Discuss the social implications of big data, data warehousing, data mining 	
Solution Development: Application Development using high-level programming language (±4 weeks/16 hours)	
Extend database and programming to incorporate relational databases <ul style="list-style-type: none"> • Accessing a relational database through a programming language Extend database and programming <ul style="list-style-type: none"> • Design and develop a solution incorporating SQL • SELECT all or some fields (FROM) and/or FORMAT • DISTINCT and ORDER BY • SELECT with calculated fields (AS) <ul style="list-style-type: none"> ◦ ROUND, INT, LENGTH ◦ LEFT, RIGHT, MID ◦ Concatenating fields • WHERE <ul style="list-style-type: none"> ◦ >, >=, <, <=, =, LIKE, BETWEEN ◦ AND, OR, NOT ◦ IS NULL, NULL, IN ◦ Wild cards ◦ #date# • INSERT, DELETE, UPDATE – one and/or multiple records • Functions and aggregate functions <ul style="list-style-type: none"> ◦ SUM, MIN, MAX, AVG, COUNT, TOP ◦ Use with GROUP BY and HAVING • Create a join query (single joins) using 'WHERE' • Mathematical operators • Date functions (day, month, year, date) • Queries with parameters where the user input is given: <ul style="list-style-type: none"> ◦ to modify data in a table ◦ to search a table ◦ implement the parameter as part of the SQL construct (only one parameter used) <ul style="list-style-type: none"> ▪ (parameter name) ▪ (parameter by name) method 	
Solution Development: Software Engineering Principles and PAT (±1½ weeks/6 hours)	
<ul style="list-style-type: none"> • Start with PAT – Task description and analysis of requirements using an appropriate methodology 	
Assessment (PoA): <ul style="list-style-type: none"> • 1 Alternative task: Closed or open book, OR case study OR integrated task • 1 Theory test 	Refer to Section 4 for: <ul style="list-style-type: none"> • Mark and time allocation • SBA weighting • Term reporting

Grade 12: Term 2 – 10 weeks/40 hours, including examinations (2 weeks)
Systems Technologies: Computer Management (±½ week/2 hours)
Factors influencing computer management (extend computer management issues from grade 10 and 11)
<ul style="list-style-type: none"> • Recommend management tasks for general housekeeping and to maintain data integrity and protect the system
Systems Technology: Software (±1 week/4 hours)
<ul style="list-style-type: none"> • Overview of cloud computing, artificial intelligence, virtual reality, augmented reality, mixed reality and virtualisation <ul style="list-style-type: none"> ○ Cloud computing <ul style="list-style-type: none"> ▪ Describe cloud computing ▪ Effect on hardware needs ▪ Software as a service (SaaS) <ul style="list-style-type: none"> – Description and advantages – Ownership of data ○ Artificial Intelligence (AI) <ul style="list-style-type: none"> ▪ Describe AI ▪ What is it used for? ▪ Advantages, Disadvantages & Limitations of AI ○ Virtual reality (VR), Augmented reality (AR), Mixed reality (MR) <ul style="list-style-type: none"> ▪ Describe VR/AR/MR ▪ What is it used for? ▪ Hardware and software requirements ▪ Advantages of VR/AR/MR ▪ Limitations of the technologies ○ Virtualisation <ul style="list-style-type: none"> ▪ Describe virtualisation ▪ What is it used for? ▪ Benefits of virtualisation
Social Implications (±½ week/2 hours)
<ul style="list-style-type: none"> • Computer criminals <ul style="list-style-type: none"> ○ Hackers (Threat actor), crackers (ethical hacker), cyber gangs, virus authors • Types of cybercrime • Effect of cybercrime • Computer crimes such as hardware, software, information, identity theft, bandwidth theft, theft of time and services <ul style="list-style-type: none"> ○ Internet-related fraud scams ○ Internet attacks (worms, malware, denial of service, back doors) ○ Phishing ○ Unauthorised remote control and administration, e.g. botnets, zombies ○ Right to access vs right to privacy, misuse of personal information • Safeguards against computer crimes, threats and criminals
Solution Development: Application Development using a high-level programming language (±4 weeks/16 hours)
<ul style="list-style-type: none"> • Object-oriented programming (OOP) <ul style="list-style-type: none"> ○ Parameterised and non-parameterised constructors ○ Correct use of private and public attributes, accessor, mutator and auxiliary methods ○ The use of the ToString method and accessor methods to provide output ○ Correct instantiation of objects ○ Correct use of methods of various objects as part of problem-solving • Arrays as a data structure (2-dimensional) <ul style="list-style-type: none"> ○ Structure ○ Populate an array (user input or hard code)

<ul style="list-style-type: none"> o Loop through items o Basic operations: row/column aggregates 	
Solution Development: Software Engineering Principles and PAT (±2 weeks/8 hours)	
<ul style="list-style-type: none"> • Reinforce software engineering principles • Interface design: Functionality and usability principles and program design • Practical Assessment Task – continue 	
Assessment (PoA): Reporting <ul style="list-style-type: none"> • 1 Practical Test • Mid-year examination <ul style="list-style-type: none"> o 1 Practical paper o 1 Theory paper 	Refer to Section 4 for: <ul style="list-style-type: none"> • Mark and time allocation • SBA weighting • Term reporting

Grade 12: Term 3 – 10 weeks/40 hours, including examinations (3 weeks)
Internet Technologies: Internet Services Technologies (±½ week/2 hours)
<ul style="list-style-type: none"> • Improve searching <ul style="list-style-type: none"> o Semantic search o Mediated search o SEO (Search Engine Optimisation) • Online applications <ul style="list-style-type: none"> o Storing data <ul style="list-style-type: none"> ▪ Locally (cookies) ▪ Online (databases) ▪ Role of SQL, scripting languages (e.g. PHP, JavaScript), XML o Running instructions <ul style="list-style-type: none"> ▪ Locally (scripts, AJAX) ▪ Online (server side, scripts and code) o Formatting output <ul style="list-style-type: none"> ▪ CSS
Communication Technologies Networks (±½ week/2 hours)
<p>Extend communication technology networks from grade 11</p> <ul style="list-style-type: none"> • Essential parts to link nodes <ul style="list-style-type: none"> o Switch, cables, wireless base station • Connecting to the Internet <ul style="list-style-type: none"> o Router/modem o Fibre, Wi-Fi, 4G/5G, Satellite • All-in-one solution ('router' is modem, router, switch and base station – all in one) • Sharing concepts <ul style="list-style-type: none"> o Sharing files and folders, user rights, BitTorrent (Risks and benefits) o Online services (Dropbox/G-Workspace/Microsoft 365) • Remote access <ul style="list-style-type: none"> o On local network, through Internet, VPN
Communication Technologies: E-communications (±½ week/2 hours)
<ul style="list-style-type: none"> • Overview of security concepts (purpose and use) <ul style="list-style-type: none"> o Review gr 11 Data Security (Blockchain, Strong passwords, encryption, multi-layer verification) o SSL (private and public key) o Certificates and security o Firewall

Social Implications (±½ week/2 hours)	
Social issues applicable to term 3 content <ul style="list-style-type: none"> • Explain how computers provide solutions to issues of national and international importance such as: <ul style="list-style-type: none"> ◦ Distributed computing power ◦ Decision making • Describe the evolution of social networking and the effect on society: <ul style="list-style-type: none"> ◦ Information overload ◦ Availability of personal information <ul style="list-style-type: none"> ▪ Consequences of search engines and group communications ▪ Social, political, environmental ▪ Global community – cultural effects ▪ Social websites and social engineering ▪ Wikis • List and discuss issues regarding privacy and information sharing <ul style="list-style-type: none"> ◦ Cookies, anonymity, Global Unique Identifiers (GUIDs), file sharing – movies, music 	
Solution Development: Application Development using a high-level programming language (±2 weeks/8 hours)	
<ul style="list-style-type: none"> • Consolidate and reinforce content, concepts and skills • Design and develop solutions for a variety of problems that include computational thinking and applying software engineering principles <ul style="list-style-type: none"> ◦ Test and validate a solution against a set of design specifications ◦ Alter a solution to meet a set of design specifications ◦ Document a solution design and development ◦ Motivate the design and development of the solution ◦ Evaluate a solution against other solutions 	
Solution Development: Software Engineering Principles and PAT (±3 weeks/12 hours)	
<ul style="list-style-type: none"> • Reinforce software engineering principles • Practical Assessment Task – Finalise 	
Assessment (PoA): <ul style="list-style-type: none"> • Prelim examination <ul style="list-style-type: none"> ◦ 1 Practical paper ◦ 1 Theory paper Practical Assessment Task (PAT): <ul style="list-style-type: none"> • To be completed before the prelim examination 	Refer to Section 4 for: <ul style="list-style-type: none"> • Mark and time allocation • SBA weighting • Term reporting

Grade 12: Term 4 – 10 weeks/40 hours, including examination (7 weeks/28 hours)
Content using Case Studies – All topics (±1½ weeks/6 hours)
Consolidate content, concepts and skills using case studies to: <ul style="list-style-type: none"> • Identify the basic hardware configuration of a computer in terms of the processor, memory and hard drive size • Understand computers and their uses • Know how to use computers as tools to access information and to communicate with others around the world • Make better buying decisions – interpret advertisements and make judgements about quality and usefulness when buying equipment and software • Know how to fix simple computer problems and deal with challenges that arise with utilising computers (and know when to call for help) • Know what kind of computer uses could benefit and advance work place and career path opportunities • Know how to protect themselves against online villains and threats • Know how to apply digital tools to communicate, gather, analyse, use information and solve problems • Understand technology concepts, systems and operations • Recommend specific hardware/software for a specific scenario

Solution Development: Application Development ($\pm 1\frac{1}{2}$ weeks/6 hours)	
Consolidate content, concepts and skills to develop a software solution	
External examination (± 7 weeks / 24 hours) <ul style="list-style-type: none"> • Practical examination (Paper 1) • Theory examination (Paper 2) • Practical Assessment Task (Paper 3) School Based Assessment (SBA) <ul style="list-style-type: none"> • Refer to Section 4 for SBA weightings 	Reporting (promotion mark) Final Practical paper – 25% Final Theory paper – 25% PAT – 25% SBA – 25%

SECTION 4

ASSESSMENT IN INFORMATION TECHNOLOGY

4.1 Introduction

Assessment is a continuous planned process of identifying, gathering and interpreting information about the performance of learners, using various forms of assessment. It involves four steps: generating and collecting evidence of achievement, evaluating this evidence, recording the findings and using this information to understand and thereby assist the learner's development to improve the process of learning and teaching.

Assessment involves activities that are undertaken throughout the year. In grades 10 – 12 assessment comprises two different but related activities: informal daily assessment (assessment for learning) and formal assessment (assessment of learning).

Assessment in IT should encourage computational thinking practices, i.e. integrating the power of human thinking with the capabilities of ICTs and computer programming.

4.2 Informal or daily assessment

Assessment for learning has the purpose of continuously collecting information on a learner's achievement that can be used to improve their learning.

Informal assessment is the daily monitoring of learners' progress. This is done through observation, discussion, practical demonstrations, learner-teacher conferences, informal classroom interactions, etc. Informal assessment may be as simple as stopping during the lesson to observe learners or to discuss with learners how learning is progressing. Informal assessment should be used to provide feedback to the learners and to inform planning for teaching but need not be recorded. It should not be seen as separate from learning activities taking place in the classroom. Learners or teachers can mark these assessment tasks.

Self-assessment and peer assessment actively involves learners in assessment. This is important as it allows learners to learn from and reflect on their own performance. The results of the informal daily assessment tasks are not formally recorded unless the teacher wishes to do so. The results of daily assessment tasks are not used for promotion and certification purposes.

4.3 Formal assessment

All assessment tasks that make up a formal programme of assessment for the year are regarded as formal assessment. Formal assessment tasks are marked and formally recorded by the teacher for progression and certification purposes. All formal assessment tasks are subject to moderation for the purpose of quality assurance and to ensure that appropriate standards are maintained.

Formal assessment provides teachers with a systematic way of evaluating how well learners are progressing in a grade and in a particular subject. Examples of formal assessments include tests, examinations, practical tasks, projects, etc. Formal assessment tasks form part of a year-long formal programme of assessment in each grade and subject.

The following tables provide the formal assessment requirements for Information Technology:

Grade 10

Formal Assessment			
During the Year	End-of-Year Examination		
40%	60%		
SBA tasks	Practical Assessment Task	End-of-Year Exam Papers (40%)	
40%	20%	20%	20%
<ul style="list-style-type: none"> 3 tests 1 alternative task/test 1 exam (mid-year) 	Project Software development project including aspects of planning cycle as well as principles of software engineering	Written exam 3 hours Theory aspects of all content, concepts and skills of all topics	Practical exam 3 hours Solution Development

Grade 11

Formal Assessment			
During the Year	End-of-Year Examination		
40%	60%		
SBA tasks	Practical Assessment Task	End-of-Year Exam Papers (40%)	
40%	20%	20%	20%
<ul style="list-style-type: none"> 3 tests 1 alternative task/test 1 exam (mid-year) 	Project Software development project including aspects of planning cycle as well as principles of software engineering	Written exam 3 hours Theory aspects of all content, concepts and skills of all topics	Practical exam 3 hours Solution Development

Grade 12

Formal Assessment			
During the Year	End-of-Year Examination		
25%	75%		
SBA	Practical Assessment Task	End-of-Year Exam Papers (50%)	
25%	25%	25%	25%
<ul style="list-style-type: none"> 2 tests 1 alternative task/test 2 exams <ul style="list-style-type: none"> Mid-year Prelim 	Project Software development project including aspects of planning cycle as well as principles of software engineering	Written exam 3 hours Theory aspects of all content, concepts and skills of all topics	Practical exam 3 hours Solution Development

The forms of assessment used should be age and developmental level appropriate. The design of these tasks should cover the content of the subject and include a variety of tasks designed to achieve the objectives of the subject.

4.3.1 Types of formal assessment for Information Technology Project

A project assesses the learner's ability to apply knowledge, skills and a range of competencies in an integrated manner, many of which cannot be assessed in other ways. It has a degree of open-endedness but is focused and results in individual but similar tasks. The time to complete a project normally ranges from a few days to several weeks.

In IT the project is the practical assessment task (PAT).

The project should enable a learner to apply a combination of techniques, knowledge and skills to new situations to complete the task or accomplish a goal. It should also encourage learners to use and combine information, data, and ideas to solve problems, discover and explain patterns, relationships or trends and predict behaviour/events.

A project should require the learner to:

- do some planning/preparation/investigation/research/data gathering to solve the identified problem/task;
- perform the task/carry out instructions (according to criteria given);
- produce a product such as a software application (this could include a limited number of smaller products such as a planning document, that builds up to the final product, which the teacher could monitor or assess informally or formally);
- demonstrate thinking and decision-making skills; and
- demonstrate some innovation and creativity.

To set and manage the project, the teacher should:

- determine the content/skills/knowledge to be addressed;
- set clear criteria and give clear instructions to guide the learner (the learner should know exactly what to do and what is expected);
- keep the scope manageable;
- determine which resources will be required to complete the project and ensure that learners have access to these resources;
- determine the time frame/duration/due date;
- determine mark distribution and compile an assessment tool; and
- continuously monitor the completion of the project and guide the learners.

Tests

A test could be a practical test or a written test. The programme of assessment should reflect a balance between practical and written tests. Tests could include open book tests.

A test for formal assessment should not comprise of a series of small tests but should cover a substantial amount of content and the duration should be a minimum of 60 minutes.

Open book tests require learners to find information and apply knowledge and skills. Learners are tested on understanding and application of learning material and not on rewriting. Open book tests should not include only short questions. They must include questions/tasks that will encourage thinking and decision making.

For written open book tests, learners are required to write longer reflective answers, such as paragraph type responses to a given scenario. Paragraphs providing reasons and supporting evidence/arguments are essential.

For practical open book tests learners are required to apply a combination of a series of procedures and techniques to new situations to provide a specific answer or accomplish a specific goal.

Alternative Assessment

Alternative assessment is an alternative to standard tests and exams. It provides a true evaluation of what the learner has learned, going beyond acquired knowledge by looking at their application of this knowledge.

Integrated task/test

An integrated task/test requires learners to be able to apply their knowledge and skills in both theory and practical work that was covered. Testing these types of scenarios e.g. database theory together with database practical, algorithm with implementation and using a trace table to debug a programme.

Case study

Case studies are investigations of real-life situation or simulation thereof. Data is gathered from a variety of sources and by using several different methods. A case study requires an in-depth, and detailed examination of a scenario, as well as the related contextual conditions.

Each test, open book test, alternative assessment task and examination must reflect different cognitive levels.

4.4 Cognitive and difficulty levels of formal assessments

Formal assessments must cater for a range of cognitive levels and abilities of learners as shown in the table below:

Cognitive Level	Taxonomy	Description
C1	Knowledge, Remembering	Recall of factual/process knowledge in isolation , i.e. one step/ set of basic steps/instruction/process at a time, e.g. definitions in the theory paper and known procedures/algorithms in the practical paper.
C2	Understanding, Applying	Demonstrates understanding of steps/algorithms/processes/ isolatable bits, such as translating from one form of representation to another, e.g. converting a flow chart representation of a program/program segment to a functional program. It also requires using known routines/algorithms/processes in a familiar context in order to complete a task, where all of the information required is immediately available to the learner .
C3	Analysing, Evaluating, Creating	Requires reasoning/investigation/developing a plan or sequence of steps/algorithm; has some complexity where candidates need to see how parts relate to a whole; organising/ putting together component parts/elements to form a coherent functional whole/achieve an overall objective and completing a task could have more than one possible approach. It could also require weighing possibilities, deciding on the most appropriate solution, as well as testing to locate errors/ troubleshooting, pattern recognition and generalisation. These questions will comprise actions/strategies/procedures where candidates are required to create their own solutions to challenges they may encounter. These questions could include analysing questions or data, and decision-making.

Levels of difficulty are categorised as follows:

- D1: Easy for the average Grade 12 candidate to answer
- D2: Moderately challenging for the average Grade 12 candidate to answer
- D3: Difficult for the average Grade 12 candidate to answer
- D4: Very difficult for the average Grade 12 candidate to answer. The skills and knowledge required to answer questions at this level should be included to distinguish amongst high achievers.

Questions in the formal assessment tasks will assess performance at different cognitive levels, critical thinking skills, problem-solving techniques and difficulty, as outlined below.

In judging the level of difficulty of each question, both the demands that each question makes on the cognitive ability of an average Grade 12 IT learner and the intrinsic level of difficulty of the question or task are considered. In making this judgement, the difficulty or ease of a particular question is identified. A four-category framework for thinking about question or item difficulty adapted from Leong (2006) has been used in this identification process. This framework comprises the following four general categories of difficulty:

- **Content difficulty:** This indexes the difficulty of the subject matter, topic or conceptual knowledge; some content is inherently more difficult than other content.
- **Stimulus difficulty:** This relates to the linguistic features of the question and the challenge that candidates face in reading, interpreting and understanding the question.
- **Task difficulty:** This refers to the difficulty that candidates face when trying to formulate or produce an answer.
- **Expected response difficulty:** This refers to difficulties because of the mark scheme or marking guidelines, in other words how marks are to be allocated.

Weighting of cognitive levels and difficulty levels

Papers 1 and 2 will include questions across three cognitive levels. The distribution of cognitive levels in the practical and theory papers is given in the table below.

Cognitive Level	Description	Paper 1 (Practical)	Paper 2 (Theory)
1	Knowledge and remembering	30%	40%
2	Understanding and applying	40%	40%
3	Analysing, evaluating and creating	30%	20%

The estimated percentages for each level of difficulty within each cognitive level are shown in the table below.

	D1	D2	D3	D4	TOTAL
C1	±10%	±10%	±10%	-	±30%
C2	±15%	±15%	±8%	±2%	±40%
C3	±15%	±7%	±5%	±3%	±30%
TOTAL	±40%	±32%	±23%	±5%	100%

Learners are required to investigate and analyse problems in a variety of contexts (such as scientific, technological, environmental and everyday-life contexts) in order to solve the described problems effectively, either via programming code in Paper 1 or describe proposed solutions in Paper 1/Paper 2.

4.4.1 Programme of Assessment

The following tables provide the programme of assessment requirements for Information Technology:

In the table below a more detailed programme of assessment:

GRADE 10								
	TERM 1	TERM 2			TERM 3		TERM 4	
	Task 1	Task 2	Task 3		Task 4	Task 5	Task 6	
Form / Types of Assessment	Test 1 Theory	Test 2 Practical	Exam Practical	Exam Theory	Alternative Task: Closed or Open Book or Case Study or Integrated test	Test 5 Practical	Final Practical Exams	Final Theory Exams
SBA Weight	15%	15%	40%		15%	15%		
Promotion Weight	40%						20%	20%
Total Marks	Minimum 45	Minimum 45	100	100	Minimum 45	Minimum 45	120	120
Time Allocation	Minimum 60 Minutes	Minimum 60 Minutes	Minimum 2.5 Hours	Minimum 2 Hours	Minimum 60 Minutes	Minimum 60 Minutes	3 Hours	3 Hours
					Promotion Weight of PAT: 20%			

GRADE 11								
	TERM 1		TERM 2		TERM 3		TERM 4	
	Task 1	Task 2	Task 3		Task 4	Task 5	Task 6	
Form / Types of Assessment	Test 1 Theory	Test 2 Practical	Exam Practical	Exam Theory	Alternative Task: Closed or Open Book or Case Study or Integrated test	Test 5 Practical	Final Practical Exams	Final Theory Exams
SBA Weight	15%	15%	40%		15%	15%		
Promotion Weight	40%						20%	20%
Total Marks	Minimum 45	Minimum 45	120	120	Minimum 45	Minimum 45	150	150
Time Allocation	Minimum 60 Minutes	Minimum 60 Minutes	3 Hours	3 Hours	Minimum 60 Minutes	Minimum 60 Minutes	3 Hours	3 Hours
			Promotion Weight of PAT: 20%					

GRADE 12									
	TERM 1		TERM 2			TERM 3		TERM 4	
	Task 1	Task 2	Task 3	Task 4		Task 5		FINAL NSC EXAM	
Form / Types of Assessment	Test 1 Theory	Alternative Task: Closed or Open Book or Case Study or Integrated test	Test 2 Practical	Exam Practical	Exam Theory	Prelim Practical Exam	Prelim Theory Exam	Final Practical Exams	Final Theory Exams
SBA Weight	6%	6%	6%	41%		41%			
Promotion Weight	25%							25%	25%
Total Marks	Minimum 45	Minimum 45	Minimum 45	150	150	150	150	150	150
Time Allocation	Minimum 60 Minutes	Minimum 60 Minutes	Minimum 60 Minutes	3 Hours	3 Hours	3 Hours	3 Hours	3 Hours	3 Hours
	Promotion Weight of PAT: 25%								

4.4.2 Examinations

Practical Assessment Task (20% - 25% of the total marks for the subject)

The IT PAT assesses the learners' ability to develop a solution for a specific task using the software development tools studied in Grades 10 – 12.

Learners should apply appropriate problem-solving techniques and software engineering principles in developing the application.

The IT PAT comprises different components/stages that represent the software development process using any appropriate approach/methodology. Software development activities typically include aspects such as:

- planning (understanding the problem/task and identifying the requirements);
- design (interface and program design using appropriate design tools and techniques – learners will not be expected to use any specific software design tool); and
- solution for the PAT (coding, testing, implementation and internal documentation.

The above can be implemented using flexible agile methodology.

In Information Technology the PAT counts 25% of the total promotion/certification mark for the subject in grade 12 (20% in grade 10 and 11). It is implemented throughout the school year and should be undertaken as one extended task, which is broken down into different phases or a series of smaller activities.

Each task must include a declaration of authenticity.

In Grade 12, the criteria for the Practical Assessment Task are externally set, internally administered and marked and externally moderated.

The Grade 12 PAT is set by the examination, moderated by Umalusi and provided to schools at the beginning of each academic year.

Paper 1: 3-hour practical paper of 150 marks (25% of the total marks for the subject)

This will be a practically oriented paper covering questions on Solution Development.

To successfully complete this paper, each learner must have access to his or her own computer in the exam room. Provision needs to be made for sufficient computers to enable the examination to be completed in **2 sittings**.

This paper assesses the practical skills as well as the knowledge and understanding underlying the skills pertaining to Solution Development, i.e. the high-level programming language studied which includes interaction with a database.

The paper does not have an overarching scenario. Each question may have its own scenario.

The paper will comprise questions covering the following broad topics:

- Basic programming skills
- Database:
 - Integrated SQL solution that will also include problem-solving as part of the solution
 - Integrated data-aware that will also include problem-solving as part of the solution
- OOP-programming (including basic application and basic problem solving skills)
- General problem solving – different levels of higher order skills

The following format to be used:

Question	Skills Tested	Marks
1	Basic, general programming skills	±45
2	Database	±40
3	Object-oriented programming (OOP)	±35
4	General problem-solving	±30

Software design tools may be provided as part of the problem statement (IPO tables, basic flow charts and class diagrams).

Learners need to apply any ONE of the TWO listed searching and sorting algorithms for practical assessment purposes.

The learner will not be required to enter large amounts of data. The required data could be retrieved from the data disk or imported from documents such as a text file, or a database table. All GUIs will be provided.

Marks for questions must be allocated towards basic skills, concepts, constructs and problem-solving techniques, e.g. application of an iteration structure as part of the solution (correct structure) as well as for the correct use of the structure. The allocation of marks should take into account the time spent on solving, coding and debugging a solution.

Paper 2: 3-hour written paper of 150 marks (25% of the total marks for the subject)

The paper will cover all theory aspects of all content, concepts and skills of topics, including elements of Solution Development, e.g. algorithmic development, data structures, program design and general programming concepts as well as generic problem-solving questions.

The paper does not have an overarching scenario. Each question may have its own scenario.

The following format to be used:

Question		Description
1	Human Computer Interaction and Social Implications These topics could be integrated as part of the other sections and will not be a separate section in the paper.	Short questions (±20 marks) A range of short questions covering all topics that could include <ul style="list-style-type: none">multiple-choice andmodified true and false.
2		Systems Technologies (±25 marks) Questions related to the content, concepts and skills in the Systems Technologies topic.
3		Communications Technologies and Network Technologies (±25 marks) Questions related to the content, concepts and skills in the Communication Technologies and Network Technologies topic.
4		Data and Information Management (±25 marks) Questions related to the management of data and the concept of information management.
5		Solution Development (±25 marks) Questions aligned to the Solution Development topic which assess the knowledge and understanding underlying the concepts and skills in the Solution Development topic.
6		Integrated Scenario (±30 marks) This section is based on a single large-scale scenario and assess all the topics.

Software tools for examination purposes as part of the theory paper are limited to IPO tables, basic flow charts, class diagrams, trace tables and use case diagrams.

Content to be covered

Assessment addresses the content as set out in this document. Due to the conceptual progression of the content across the grades, content and skills from Grades 10 – 12 will be assessed in the external papers at the end of Grade 12.

Every three years content will be revisited to include new and emerging technologies and trends and remove obsolete technologies and trends.

4.5 Recording and reporting

Recording is a process in which the teacher documents the level of a learner's performance in a specific assessment task. It indicates learner progress towards the achievement of the knowledge as prescribed in the Curriculum and Assessment Policy Statements. Records of learner performance should provide evidence of the learner's conceptual progression within a grade and her or his readiness to progress or be promoted to the next grade. Records of learner performance should also be used to verify the progress made by teachers and learners in the teaching and learning process.

Reporting is a process of communicating learner performance to learners, parents, schools, and other stakeholders. Learner performance can be reported in a number of ways. These include report cards, parents' meetings, school visitations, parent-teacher conferences, phone calls, letters, class or school newsletters, etc. Teachers in all grades report in percentages against the subject.

Seven levels of competence have been described for each subject listed for Grades R – 12. The various achievement levels and their corresponding percentage bands are as shown in the table below:

Codes and percentages for recording and reporting

Rating Code	Description of Competence	Percentage
7	Outstanding achievement	80 – 100
6	Meritorious achievement	70 – 79
5	Substantial achievement	60 – 69
4	Adequate achievement	50 – 59
3	Moderate achievement	40 – 49
2	Elementary achievement	30 – 39
1	Not achieved	0 – 29

Teachers will record actual marks against the task by using a record sheet; and report percentages against the subject on the learners' report cards.

4.6 Moderation of assessment

Moderation refers to the process that ensures that the assessment tasks are fair, valid and reliable. Comprehensive and appropriate moderation practices must be in place for the quality assurance of all subject assessments.

4.6.1 Formal assessment (SBA)

- Grade 10 and 11 tests and examinations are internally moderated. The subject advisor must moderate a sample of these tasks during his/her school visits to verify the standard of tasks and the internal moderation.
- Grade 12 tests and examinations must be moderated at provincial level. This process will be managed by the provincial education department.
- Subject advisors must moderate samples of tests and examination papers before they are written by learners to verify standards and guide teachers on the setting of these tasks.

4.6.2 Practical Assessment Task (PAT)

- Grade 10 and 11: Teachers assess the practical assessment tasks in Grades 10 and 11. The subject advisor must moderate a sample of PATs during his/her school visits to verify the standard of tasks and the internal moderation
- Grade 12: Teachers assess the practical assessment tasks according to the externally set

assessment tool. The subject advisor must moderate a sample of each phase of the PATs during his/her school visits to verify the interpretation of the assessment tool and the standard of marking. Completed PATs must also be moderated at provincial level. This process will be managed by the provincial education department.

4.7 General

This document should be read in conjunction with:

4.4.3 *National policy pertaining to the programme and promotion requirements of the National Curriculum Statement Grades R – 12; and*

4.4.4 The policy document, *National Protocol for Assessment Grades R – 12*.

4.8 Annexures

Annexure A – Glossary of acronyms and abbreviations

Annexure B – Glossary of components, events and methods

Annexure C – A taxonomy description for practical content

ANNEXURE A

GLOSSARY OF ACRONYMS AND ABBREVIATIONS

1:M	One-to-many
1-D	One-dimensional (array)
2-D	Two-dimensional (array)
3-D (printer)	Three Dimensional (printer)
3G	Third generation of cellular wireless
4G	Fourth generation of cellular wireless
5G	Fifth Generation of cellular wireless
4IR	Fourth Industrial Revolution
ADSL	Asymmetric Digital Subscriber Line
AI	Artificial Intelligence
AJAX	Asynchronous JavaScript and XML
ASCII	American Standard Code for Information Interchange
BIOS	Basic Input Output System
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DBA	Database Administrator
DBMS	Database Management System
DIV	Integer division
DVD	Digital Versatile Disc
EDP	Event Driven Programming
ERD	Entity Relationship Diagrams
FOSS	Free Open-Source Software
FTP	File Transfer Protocol
GIGO	Garbage-In Garbage-Out
GPS	Global Positioning System
GPU	Graphic Processing Unit
GUI	Graphical User Interface
HAN	Home Area Network
HCI	Human Computer Interface
HDD	Hard Disk Drive
HDMI	High-Definition Multimedia Interface
HTML	Hypertext Mark-up Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
I/O	Input-Output
ICT	Information and Communication Technology
IDE	Integrated Development Environment

IMAP	Internet Message Access Protocol
IoT	Internet of Things
IP	Internet Protocol
IPO	Input-Processing-Output
IPTV	Internet Protocol Television
ISP	Internet Service Provider
IT	Information Technology
JPEG	Joint Photographic Experts Group
LAN	Local Area Network
LCD	Liquid Crystal Display
LED	Light Emitted Diode
LTE	Long-Term Evolution
MOD	Modulus
MP3	MPEG-1 Audio Layer-3
MPEG	Motion Picture Expert Group
NFC	Near Field Communication
NIC	Network Interface Card
NOS	Network Operating System
OOP	Object-Oriented Programming
OS	Operating System
OTP	One Time Pin
PAN	Personal Area Network
PAT	Practical Assessment Task
PC	Personal Computer
PHP	Hypertext Preprocessor
PnP	Plug-and-Play
PoA	Programme of Assessment
POP3	Post Office Protocol
PoS	Point-of-Sale
RAD	Rapid Application Development
RAM	Random Access Memory
RFID	Radio-Frequency Identification
ROM	Read-Only Memory
SaaS	Software as a Service
SEO	Search Engine Optimisation
SMS	Short Message System
SMTP	Simple Mail Transfer Protocol
SOHO	Small Office Home Office
SQL	Structured Query Language
SSD	Solid State Drive
SSL	Secure Socket Layer

TOE	Task-Object-Event
UML	Unified Modelling Language
UPS	Universal Power Supply
URL	Uniform Resource Locator
USB	Universal Serial Bus
UTF	Unicode Transformation Format
VGA	Video Graphic Array
VOD	Video On Demand
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
VRAM	Video Random Access Memory
W3C	World Wide Web Consortium
WAN	Wide Area Network
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WWW	World Wide Web
WYSIWIG	What You See Is What You Get
XML	Extensible Markup Language

ANNEXURE B

	Component, Event, Method	10	11	12
Components	TForm	√	√	√
	TPageControl	√	√	√
	TButton	√	√	√
	TBitButton	√	√	√
	TLabel	√	√	√
	TEdit	√	√	√
	TImage	√	√	√
	TShape	√	√	√
	TPanel	√	√	√
	TListBox	√	√	√
	TRadioGroup	√	√	√
	TComboBox	√	√	√
	TMemo	√	√	√
	TRadioButton	√	√	√
	TCheckBox	√	√	√
	TSpinEdit	√	√	√
	TTimer	√	√	√
	TStringGrid		√	√
	TDBGrid		√	√
	TADOTable		√	√
	TADOQuery		√	√
	TDataSource		√	√
	InputBox()	√	√	√
	ShowMessage()	√	√	√
	MessageDlg()	√	√	√
Events	OnClick()	√	√	√
	OnCreate()	√	√	√
	OnActivate()		√	√
	OnShow()		√	√
	OnClose()		√	√
	OnTimer	√	√	√

	Component, Event, Method	10	11	12
Methods - Functions - Procedures	Show()	√	√	√
	Hide()	√	√	√
	Length()	√	√	√
	setLength		√	√
	SetFocus()	√	√	√
	IN operator	√	√	√
	Pos()		√	√
	Copy()		√	√
	Insert()		√	√
	Delete()		√	√
	Concat()		√	√
	AssignFile()		√	√
	Append ()		√	√
	Reset()		√	√
	Rewrite()		√	√
	CloseFile ()		√	√
	Ord()		√	√
	Chr()		√	√
	Val()		√	√
	Str()		√	√
	UpCase()	√	√	√
	UpperCase()	√	√	√
	LowerCase ()		√	√
	IgnoreCase ()		√	√
	FileExists()		√	√
	ReadLn()		√	√
	WriteLn()		√	√
	LoadFromFile()	√		
	SaveToFile()	√		

	Component, Event, Method	10	11	12
Methods - Functions - Procedures	Conversion / Formatting			
	IntToStr () / StrToInt ()	√	√	√
	FloatToStr () / StrToFloat ()	√	√	√
	FloatToStrF ()	√	√	√
	DateTime Functions		√	√
	FormatDateTime()		√	√
	TimeToStr()		√	√
	DateToStr()		√	√
	DateTimeToStr()		√	√
	StrToDate()		√	√
	StrToTime()		√	√
	Now()		√	√
	Date()		√	√
	Time()		√	√
	IsLeapYear()		√	√
	Mathematical Methods			
	Random()	√	√	√
	RandomRange()		√	√
	Round()	√	√	√
	Trunc()	√	√	√
	Frac()		√	√
	Ceil()		√	√
	Floor()		√	√
	Sqr()	√	√	√
	Sqrt()	√	√	√
	Inc()		√	√
	Dec()		√	√
	PI		√	√
	Power()		√	√

ANNEXURE C

A TAXONOMY DESCRIPTION FOR PRACTICAL CONTENT

Lower Order (C1)	Middle Order (C2)	Higher Order (C3)
30%	40%	30%
Knowledge/Remembering	Understanding/applying	Analysing/evaluating/creating
<p>Code Generator</p> <p>Operates at level of individual lines of code/code structures / routine procedures (in isolation)</p> <p>The learner is able to</p> <ul style="list-style-type: none"> recall specific isolatable bits of information learned use bits of code/code structures in isolation - no real connections – in an unrelated way generate code - knows syntax and semantics - can write a line of code/a code structure that does something specific, e.g. basic processing statement, lines of code to obtain input or produce output, algorithm to swap two items, etc. to focus on one relevant aspect at a time (uni-structural) answer questions, seen before, used in exactly same context as learned/classroom-based exercise and that is straight forward, to-the-point, that requires mostly one, direct answer/piece of code/code structure <p>Cannot</p> <ul style="list-style-type: none"> See relationships Combine concepts/various lines of code/code structures to achieve a goal or complete a task <p>Examples of isolatable bits of content learned (knowledge) that the learner is able to recall and use in isolation:</p> <ul style="list-style-type: none"> syntax rule code statement, e.g. assign statement built-in method, e.g. random structure, e.g. class definition algorithm, e.g. swap two values, sort process, e.g. reading a text file, populate array setting property value 	<p>Program Generator</p> <p>Operates at level of writing basic programs that combine concepts / structures, isolatable bits</p> <p>The learner is able to</p> <ul style="list-style-type: none"> read a program, tell what each line means/ does tell the goal/outcome of a program write programs seen before in a similar context/to perform specific tasks Able to relate, combine and integrate some concepts/code/code structures into valid programs - use and combine specific building blocks to write a program for a specific task Can answer closed/scaffolded questions in a similar context than experienced before, with or without new elements <p>Cannot</p> <ul style="list-style-type: none"> Optimise a program/code Do detailed planning Perform error catching/trace errors Answer unfamiliar, unseen or open-ended questions without scaffolding and guidance <p>Understanding:</p> <p>Convert from one format to another, e.g. interpret flow chart and convert to code, Read code and tell what it does or provide the output</p> <p>Applying:</p> <p>Carrying out or using a procedure/algorithm/structure/ code statement in a given situation similar context (but <i>new elements or situation</i>) as was experienced before to perform a task, e.g. combine concepts/isolatable bits</p>	<p>Software Developer</p> <p>Operates at a level of writing solutions to new / unfamiliar or open-ended problems</p> <p>The learner is able to</p> <ul style="list-style-type: none"> tell what the different parts of a program do and how different parts of a program work together optimise a program/section of code analyse, design, plan, implement and test a solution to a new problem Perform error catching, understanding when, where and how relate, combine and integrate several code structures/constructs to devise 'new' algorithms/ adapting existing ones link several aspects to a broader context independently identify patterns and relate these to programming constructs/structures generalise, abstract and decompose problems into sub-problems and modules answer free response/open-ended questions, 'new' (unseen questions), by framing the question and finding a plausible cause of action <p>Synthesis</p> <p>Combine concepts in unfamiliar/new context to form a (new coherent or functional whole), e.g. code a solution to a problem/to perform a task (not seen before)</p> <p>Includes analysing, e.g. identifying different parts such as sub-routines/ modules/data structures / I/O strategies /algorithms required; Includes evaluating, e.g. deciding which structures to use and free response/open-ended questions</p>