basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

INFORMATION TECHNOLOGY

EXAMINATION GUIDELINES

GRADE 12

2021

These guidelines consist of 12 pages.

TABLE OF CONTENTS

1. INTRODUCTION

The Curriculum and Assessment Policy Statement (CAPS) for Information Technology outlines the nature and purpose of the subject Information Technology. This guides the philosophy underlying the teaching and assessment of the subject in Grade 12.

The purpose of these Examination Guidelines is to:

- Provide clarity on the depth and scope of the content to be assessed in the Grade 12 National Senior Certificate Examination in Information Technology.
- Assist teachers to adequately prepare learners for the examinations.

This document deals with the final Grade 12 external examinations. It does not deal in any depth with the School-Based Assessment (SBA), Practical Assessment Tasks (PATs) or final external practical examinations, as these are clarified in a separate PAT document which is updated annually.

These Examination Guidelines should be read in conjunction with:

- The *National Curriculum Statement (NCS) Curriculum and Assessment Policy Statement (CAPS)*: Information Technology
- The National Protocol of Assessment: *An addendum to the policy document, the National Senior Certificate: A qualification at Level 4 on the National Qualifications Framework (NQF), regarding the National Protocol for Assessment (Grades R–12)*
- The national policy pertaining to the programme and promotion requirements of the National Curriculum Statement, Grades R–12

2. COMPUTER LAB REQUIREMENTS

Refer to the CAPS document for the resources required for offering Information Technology, with respect to the infrastructure, equipment and finances, which are the responsibility of the school.

NOTE: Circular S9 of 2015 states, 'As from November 2017, the DBE will only use Object Pascal (Delphi) for assessment in the NSC examinations.'

Delphi 2010 is provided for free to all public schools and learners, and can be ordered from Embarcadero Technologies. Contact your IT subject advisor for the order information.

Schools must make sure that the required **programming language is installed** on all computers **at the beginning of the year**. This is to ensure that Grade 12 candidates have enough time to familiarise themselves with the software they are required to use during the final practical Information Technology (IT) examination.

2.1 Software requirements

The following software versions are required for the NSC IT examination:

- Embarcadero Delphi 2010
- Microsoft Office Access 2007 or above

If another version of the required software is used at school, note the following:

NOTE:
- The learner files for the NSC IT examination will be prepared using Delphi 2010. Schools using lower versions of Delphi (Delphi 7.0/Delphi Light/Turbo Delphi) must be aware of the error message that will display when the Delphi 2010 programs provided for the NSC examination are executed and how to fix the error.
- If a later version of Delphi is used, it is the responsibility of the IT teacher/school to ensure that the software is compatible with the data files prepared in Delphi 2010.

Also note the following:

NOTE: The Microsoft Office Access database engine 2007 or higher (English) version must be installed in order to use the '**Microsoft Office 12.0 Access Database Engine OLE DB Driver**' with the database aware components. This is in addition to the 'Microsoft Jet 4.0 OLEDB Provider' driver, which is used for older versions of Microsoft databases.

NOTE: The configuration of software mentioned above is used to prepare the learner files for the practical paper. The computer lab that will be used during the Grade 12 examination should be prepared using the same configuration.

2.2 Official checklist for the external examination session

According to examination instruction Circular E14 of 2013, each computer lab used for the Grade 12 final IT practical examination must be audited. The audit must be done using a checklist, which is supplied annually by the DBE/provincial education departments. A certificate of readiness must be submitted to the DBE after being co-signed by the IT teacher, the IT technician and the school principal, and then verified by the district facilitator.

3. ASSESSMENT

3.1 Scheme of external assessment

All candidates will write TWO external papers as prescribed.

3.2 Format of question papers

PAPER	TYPE OF PAPER	DURATION	TOTAL	MARKING
1	**Practical**	3 hours	150	External
2	**Theory**	3 hours	150	External

NOTE: Learners will receive 30 minutes reading and planning time before commencement of the practical examination.

Questions in both Papers 1 and 2 will assess performance at different cognitive levels, critical thinking skills, problem-solving techniques and difficulty, as outlined 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 simple procedures 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 that could have more than one possible approach. It could also require weighing possibilities, deciding on the most appropriate solution, and testing to locate errors/troubleshooting as well as pattern recognition and generalisation. These questions will comprise actions/strategies/procedures where candidates are required to create their own solutions to challenges different to those learners may have encountered in the classroom. 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.

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 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. Therefore, answers to multiple-choice questions on a specific topic could be easier than questions where a candidate has to construct a response.

3.3 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%	30%
2	Understanding and applying	40%	40%
3	Analysing, evaluating and creating	30%	30%

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. ELABORATION OF THE CONTENT

4.1 Paper 1: Practical paper

This question paper will require the use of a computer and will contain questions based on software solution development. This question paper assesses **practical skills** as well as the knowledge and understanding underlying programming skills pertaining to software solution development and problem-solving using Delphi and/or SQL as the high-level programming languages studied.

The following topics could be examined in the practical question paper:

- Sound programming principles
- Use of data of all data types specified in the IT CAPS document
- Application of programming statements specified in the IT CAPS document, such as the selection structures (e.g. if-statement) and repetition structures (e.g. for-statement)
- Use of GUI components as stated in the IT CAPS document and this guideline document. Refer to the table in section 4.1(i) of this document for the descriptions of components to be used in the question papers.
- Arrays: one-dimensional and two-dimensional arrays
- Object-oriented programming (OOP)
- Database: SQL and database programming using code constructs
- String manipulation
- Calculations
- Pattern recognition
- Text files for use as input/output structures
- Recall and application of all algorithms as stated in the CAPS. Refer to 4.1(d).
- All other structures and utility classes as stated in the IT CAPS document

(a) Physical requirements

- To successfully complete the practical question paper, each learner must have access to his/her own computer in the computer lab/examination room.
- The software set out in Section 2 of this document should be installed on the computers that will be used during the examination session well in advance of the examination date.

(b) Mark allocation for required skills

SECTION	SKILLS TESTED	MARKS
A	Basic, general programming skills	40 (±10)
B	Database	40 (±10)
C	Object-oriented programming (OOP)	40 (±10)
D	General problem-solving	30 (±10)

(c) Specific requirements

Graphical user interface (GUI) (included in ALL questions)

- GUIs will be provided in most cases; however, for simple/basic programs where only a form/frame and a few components are required learners could be expected to create the GUI themselves.
- Learners are expected to add/delete/modify components of GUIs provided.

Object-oriented programming (OOP)

Learners must be able to design and develop solutions for specific problems that include computational thinking and applying software engineering principles using event-driven programming within the OOP paradigm.

Learners must also pay attention to:

- 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

NOTE: The programming skills listed as basic programming skills and problem-solving skills could also be required to be applied in the OOP question.

Database

Learners must be able to manipulate a normalised relational database using data-aware components and code constructs from Delphi (with or without SQL) to do the following:

- Perform CRUD operations (create/add, read, update, delete) on records
  NOTE: Deletion of records in a relational database will only be required to perform on one table (the table that contains the foreign key).
- Manipulate data retrieved from the database, e.g. calculate VAT on prices, average, minimum, maximum and other calculations
- Text manipulation, e.g. creating passwords from fields, such as name, surname, ID
- Solve problems using data retrieved from the database that could include algorithms as listed in the IT CAPS document
- Use date functions including extracting the day, month and year from the system date
- Perform queries on the database/extract records meeting specific condition(s)
- Pre-coded methods such as filter, sort and locate will **NOT be allowed** as coding in this regard must be done from first principles
- Manipulate data from a database with or without a data module
- Solve problems using a database

The following SQL-statements could be assessed:

- Select, distinct
- Where
- Order by
- Group by
- Special operators: Between, In, Like, Is Null, Having
- Insert, Update, Delete
- Subqueries (simple form, single select or aggregate)
- Aggregate functions (sum, average, min, max, count)
- Date functions (day, month, year, date)
- String functions (length, left, right, mid, concatenation of fields)
- Dynamic queries using user input
- Queries with parameters where the user input is given to modify data in a table or to search a table
- Calculating new fields

NOTE:

- The **database connection will be provided**. A folder will be supplied that contains the project/ program that has already been connected to the database.
- A normalised relational database which contains one or more tables will be provided.
- If the database connection is NOT successful, learners must still provide the code for the given problem.

(d) Algorithms

Learners must know ALL algorithms studied and as listed in the CAPS.

<u>Examples</u> of algorithms/concepts that learners must be able to recall and use (not restricted to the list only):

- Swapping values
- Determine/Calculate the highest, lowest, average, mean, percentage increase/decrease, etc. of a number of values
- Basic calculations, such as calculating area, volume, VAT, discount
- Determine whether a number is even or odd, prime, composite, etc.
- Determine whether a number is a factor/multiple of another number
- Isolate digits in an integer number
- Determine the lowest common multiple (LCM) and greatest common divisor (GCD)
- Determine the current age based on a given date of birth or ID number
- General string manipulation, e.g. use an ID number to determine the age and gender, count vowels/words, identify palindromes
- Apply basic input and processing validation techniques, e.g. test for division by zero
- Convert a decimal number to a binary number, and vice versa
- Search for a specific element/value in an array/table with/without a flag
- Sort elements in an array
- Manipulate elements in an array

Learners must also be able to customise algorithms studied to complete tasks and to solve problems.

(e) Possible types of questions

- Convert pseudocode/flowchart representation of a program/program segment to a functional program
- Convert a UML class diagram into a class definition
- Modify/Correct/Complete a program/program segment
- Open-ended, problem-solving questions
- Structured questions

(f) General problem-solving – open-ended questions

Problem-solving questions will be of an open-ended nature where the learner must be able to apply all the programming constructs, techniques, algorithms and skills studied during Grades 10, 11 and 12 in Information Technology.

Learners must also be able to recognise patterns and customise known algorithms to suit new circumstances, e.g. convert a decimal number to a hexadecimal number (or any other base).

Design and develop solutions for problems that include computational thinking and applying software engineering principles to solve problems.

NOTE: All solutions must be developed from first principles, using algorithms. This includes routines such as search/sort and reading/writing using text files. Marks will not be allocated for the use of predefined classes/methods to perform these tasks.

(g) Excluded for Paper 1

The following concepts will not be examinable in the practical question paper, but learners may use these concepts/constructs in their PAT:

- Arrays as an attribute of an object
- Array of objects
- Inheritance and polymorphism as tools in OOP programming
- Learners will not be required to enter large amounts of data. The required data could be retrieved from the database/text files.

(h) Learner files for Paper 1: Practical paper

Learner files will be supplied which **may** include:

- GUI(s), complete or incomplete
- Text file(s)
- Object class (complete or incomplete)
- Database

(i) Reference to GUI components, events and methods in Paper 1

Refer to ANNEXURE B, page 112 of the Grade 12 Abridged Section 4 CAPS document for the components, events and methods learners will be required to use.

4.2 Paper 2: Theory paper

The question paper will include all theory aspects of all content, concepts and skills of topics, as well as elements of software solution development, e.g. algorithmic development, data structures, program design and general programming concepts, as well as generic problem-solving questions. SQL statements will be included either in the theory paper or practical paper.

Questions will be set on work taught from Grade 10 to 12, as indicated in the IT CAPS document.

Approximately 10 marks will be used to assess the latest developments in Information Technology. A few topics/concepts that will be focused on will be announced annually. On these topics, learners must be able to:

- Define/Describe the topic/concept
- Explain its function/use/role
- List advantages and/or disadvantages
- List/Describe its basic features

Advanced technical detail will not be expected for these topics/concepts.

(a) Layout of Paper 2

The following format could be used:

SECTION		DESCRIPTION
A	**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 question paper.	**Short questions** (±20 marks) A range of short questions covering all topics that could include multiple-choice and modified true/false items
B		**Systems Technologies** (±25 marks) Questions related to the content, concepts and skills in the Systems Technologies topic
C		**Communications and Network Technologies** (±25 marks) Questions related to the content, concepts and skills in the Communication Technologies and Network Technologies topic (including the internet)
D		**Data and Information Management** (±25 marks) Questions related to the management of data and the concept of information management
E		**Solution Development** (±25 marks) Questions aligned to the Solution Development topic which assesses the knowledge and understanding underlying the concepts and skills in the Solution Development topic Software design 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
F		**Integrated Scenario** (±30 marks) This section is based on a scenario and assesses topics from all sections (B to E).

The questions in SECTION E of Paper 2 (Solution Development) will be specific to the Delphi programming language. Learners may be required to write out the solution of a given problem in terms of an algorithm.

(b) General guidelines to complete Paper 2

- The mark allocation per question indicates the number of facts required.
- Avoid general answers, such as 'cheap', 'fast'.
- If two facts are asked and the learner supplies more than two facts, only the first two facts supplied will be marked.
- Do not leave answers blank. Always try to formulate an answer.

5. PRACTICAL ASSESSMENT TASK (PAT)

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)
- Coding, testing, implementation and internal documentation

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

6. CONCLUSION

It is envisaged that this Examination Guidelines document will serve as an instrument to strengthen and empower teachers to set valid and reliable assessment items in all their classroom activities.

This Examination Guidelines document is meant to articulate the assessment aspirations espoused in the CAPS document. It is therefore not a substitute for the CAPS document which teachers should teach to.

Qualitative curriculum coverage as enunciated in the CAPS cannot be over-emphasised.