education

Department:
Education
REPUBLIC OF SOUTH AFRICA

INFORMATION TECHNOLOGY

EXAMINATION GUIDELINES

GRADE 12

2009

This guideline consists of 8 pages.

Copyright reserved

1. GENERAL

1.1 National Department Documents

The 2009 IT examinations are based on the following documents:

- The National Curriculum Statement (NCS) for Information Technology (IT) (2008)
- The Learning Programme Guidelines (LPG) for IT (2008)
- The Subject Assessment Guidelines (SAG) for IT (2008)

1.2 Official Notices and Circulars

IT teachers and invigilators should be familiar with the latest official circulars pertaining to the conduct of a practical computer-based exam and the security procedures that should be in place.

Refer specifically to Annexure G (Practical examination in Computer Applications Technology and Information Technology page 75 in Volume 518 ***Government Gazette No. 31337*** of 29 August 2008 which can be found on web portal: http://www.thutong.org.za or http://www.education.gov.za .

1.3 Exemplars

The exemplar papers of 2008 as well as the November 2008 and March 2009 papers should be used as a guideline indicating the structure of the papers as well as the type of questions learners can expect in the final examination papers.

Practical Examination 2009

The practical examination paper will consist of two sections:

SECTION A for Delphi programmers
SECTION B for the Java programmers

Time allocated: 3 hours
Total marks: 120

Copyright reserved

Distribution of marks in the practical question paper (PAPER 1):

Question	Topic	Marks
One	Database Connectivity and SQL: The layout of this question will be as follows: Screenshots of the tables will be given as addendums The instructions to do the connection to the database in Delphi will be given as an addendum. Java learners will again receive the Java code to connect to the database electronically. The number of questions that make up QUESTION 1 will be restricted to a maximum of 6 questions.	**40**
Two	Object-Oriented programming: Learners must be able to create one or two classes and develop a test class (main unit) to instantiate objects and then call methods from the class/classes. Learners must be able to use structures such as text files, arrays and **array of objects**. Techniques such as searching with/without a flag, sorting, get the highest/lowest/average/total, remove/add elements from/to an array, display all/certain information could be asked. The structure will be very much the same as QUESTION 2 in the November 2008 question paper.	**±45**
Three	General programming skills and techniques will be tested: This question may contain structures such as text files, one or two dimensional arrays. Techniques such as searching with/without a flag, sorting, get the highest/lowest/average/total, remove/add elements from/to an array, display all/certain information could be asked. Learners can either create objects to answer this question or not. ANY of the concepts mentioned above can be tested in the question. The same concepts will not necessarily be tested in this question each year.	**±35**
	TOTAL:	**120**

NOTE: An important feature of programming today is modular programming (in all programming languages). This means that the emphasis is placed on objects and methods (procedures and functions). The same trend has been followed in the setting of the practical papers – specifically in QUESTION 2. If the learners cannot use subprograms/methods in their programming in general, then they will struggle to do well in the final practical examination paper.

Flow of the paper:

A scenario will be described. All the questions in the paper will be related to the scenario.

QUESTION 1: DATABASE (SQL) PROGRAMMING (40 marks)

In this question the learner will be required to manipulate the data in a database file using SQL statements.

- A specific scenario will be described that this question will be based on.
- The following will be given:
  - An incomplete program that establishes a connection with an existing database file. The entire program framework will be given (i.e. the database connection as well as the menu in Java and buttons in Delphi).
  - The database that will consist of at least two related tables populated with data.
  - Text files that can be used to create the database file with the tables if necessary.

  Java: Code to connect with the database will be supplied.
  Delphi: Connectivity will be in place. Instruction on how to connect to the database will be included in the paper.

  The learners only have to type in the code containing SQL statements to execute the various queries and display the results.

  Learners are expected to write both static (i.e. fixed) SQL and dynamic (i.e. the SQL statement is generated interactively based on user input) SQL statements.

  In 'dynamic' SQL the SQL is created using string handling to create a SQL statement based on input from the user. The user input is usually used in the ***where*** or ***order*** clauses of the SQL statement.

The SQL commands/concepts that the learners will use include:

- **SELECT**
  e.g. SELECT * FROM aDatabaseTb

- **DISTINCT**
  e.g. SELECT DISTINCT City FROM Customers
  (This will produce a list of unique city names – i.e. if a city name appears more than once in the database it will still only appear once in the list generated by this SQL statement.)

- **WHERE** (including using ***WHERE*** to link related tables as well as using 'like', 'between', 'in', 'is null' and '%')
  Examples:
  ```
  SELECT * FROM aDatabaseTb WHERE SurNameField = 'Du Plessis'
  SELECT * FROM aDatabaseTb WHERE SurNameField LIKE 'Du %'
  SELECT CustName,OderDate,OrderTotal
  FROM ClientTb, OrderTb
  WHERE OrderTb.CustID = ClientTb.CustID
  ```

- **ORDER BY** (for sorting)
- **GROUP BY** (for keeping related data together)
- Creating **calculated fields** (like calculating a mark)
  SELECT LearnerName, Subname, ***(Term1Mark + Term2Mark + Term3Mark) / 3 AS PromMark***
  FROM LearnerTb, MarkTb
  WHERE MarkTb.LearnerID = LearnerTb.LearnerID

  (This generates a field called PromMark which holds a calculated value of the three term marks added together and divided by 3.)
- **Formatting with ROUND, Int etc**
  Select LearnerName, Subname, ***Round ((Term1Mark + Term2Mark + Term3Mark) / 3, 2) as PromMark***
  FROM LearnerTb, MarkTb
  WHERE MarkTb.LearnerID = LearnerTb.LearnerID

  (This generates a field called PromMark which holds a calculated value of the three term marks added together and divided by 3 – and rounds the answer off to 2 decimal places.)
- **UPDATE & SET**
  UPDATE MarksTb SET Term1Mark = Term1Mark + 5
  WHERE SubName = 'English' AND Grd = 11

  (This statement adds 5 marks to every Grade 11 learner that has English as a subject.)
- **Summative functions**
  SUM, AVERAGE, COUNT, etc.
- **INSERT**
  To add records to a table
- **DELETE**
  To delete data from a table (usually used with a ***WHERE*** clause)
- **Date functions**
  YEAR, MONTH, DAY, etc

NOTE: Learners will **NOT** be required to alter the content of the database or create any forms, queries or reports using any software package.

Copyright reserved

QUESTION 2: OBJECT ORIENTED PROGRAMMING

A specific scenario will be described for which a program will need to be written. The question will be divided into subsections. Each subsection of the question will form a segment of the program, similar to the structure of Question 2 in the 2008 exemplar paper as well as the November 2008 practical paper.

NOTE:

- Learners should work through the paper sequentially and answer every subsection of the program to the best of their ability.
- Even if the final program does not execute correctly, marks will be awarded for all the codes.
- If learners have code that does not comply, it should not be removed but rather placed in comments. Code in comments ***will*** be marked!

Essential concepts:

- **Class definition**
  - Ensure correct use of private, protected & public
  - Parameterised and non-parameterised constructors
  - Accessor ('get' - functions) & mutator ('set' - procedures) methods
  - toString methods to produce output
  - NO INHERITANCE or POLYMORFISM for 2009
- **Arrays**
  - One dimensional array/ array of objects
  - Tracking the number of elements populating the array (as opposed to its size) using an integer variable as a counter
  - Inserting objects/primitive elements into the array
  - Deleting objects/primitive elements from the array
  - Sorting the array
  - Processing the array (i.e. summing, averaging, selecting, listing, etc.)
- **Text files**
  - Linking file handle to a text file on disk
  - Opening and closing the text file
  - Looping through the text file
  - Importing data from a text file (including reading lines from a text file)
  - Exporting data to a text file (including writing lines to a text file)
- **General concepts**
  - String handling
    - Building strings (concatenation)
    - Encryption/Decryption
    - Counting words
    - Removing items
    - Parsing items from a string
    - Changing case
    - Formatting of decimal values (i.e. limiting no of decimal places, etc)
    - Formatting of output using <tab> and <newline / return> characters / using the printf function (in java)
  - Basic mathematical manipulation
  - Looping (all 3 types)

- Decisions - if, switch/case
- Converting types where language appropriate

- **GUI components (for Delphi/Netbeans/JBuilder)**
  - Menus
  - Buttons
  - Output components such as RichEdit/Text Pane/Label/Panel/ShowMessage box/Message box
  - Input components such as Input box

QUESTION 3: GENERAL

The same specifications mentioned for QUESTION 2 are applicable to this question except for the coding of classes and objects that will be optional in QUESTION 3. The same aspects tested in QUESTION 3 in the 2008 practical paper will not necessarily be tested in the 2009 practical paper.

NOTE: Java learners can use any IDE.

Theoretical Examination 2009

The theory paper will have the same structure as that of the 2008 exemplar and the 2008 November theory papers.
A scenario will be given. Each section will consist of questions related to the overall scenario.

Time allocated: 3 hours
Total marks: 180

Distribution of marks in the theory question paper:

Section	Topic	Marks
A	Multiple-choice questions on all the assessment standards	10
B	*Hardware and software:* Covers LO 1 All the ASs in Grade 10, 11 and 12	50 - 60
C	*e-Communication, Social And Ethical Issues:* Covers LO 2 and 3 All the ASs in Grade 10, 11 and 12	15 - 20
D	*Programming and Software Development:* Covers LO 4 All the ASs in grade 10, 11 and 12	45 - 50
E	*Integrated Scenario*: Covers the LO 1, 2, and 3 as indicated in sections B and C	45 - 50

Copyright reserved

GENERAL REMARKS:

- The focus will be on the Grade 11 and 12 work as listed in the LPG (2008). Questions on some basic concepts from the Grade 10 work will be included.
- None of the questions in SECTION D (LO 4) will be specific to a programming language. Learners may be required to write out the solution of a given problem in terms of an algorithm. The marks allocated in SECTION D will vary from 1 to 12 marks per question.
- Marks allocated in SECTIONS A, B, C and E will vary from 1 to 6 marks per question. NB: The number of marks allocated to a question indicates the number of facts that should be given in the answer.
- The following topics will be covered in SECTION D of the question paper:
  - Database planning and structuring including the concept of normalization
  - Analysis of output
  - Object oriented concepts and structures excluding inheritance and polimorfism
  - Design of algorithms as the solution to a problem
  - Well-designed input screens and navigation

No questions will be asked on spreadsheets in the theory or the practical paper.

Good luck with the preparations for the final examination.

Copyright reserved