

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

SEPTEMBER 2016

INFORMATION TECHNOLOGY P1

MARKS: 150

TIME: 3 hours



This question paper consists of 10 pages.

INSTRUCTIONS AND INFORMATION

1. This question paper is divided into THREE sections. Learners must answer ALL THREE sections.
2. The duration of this examination is THREE hours. Because of the nature of this examination it is important to note that you will not be permitted to leave the examination room before the end of the examination session.
3. Make sure that you answer the questions in accordance to the specifications that are given in each question. Marks will be awarded according to the set requirements.
4. Answer only what is asked in each question. For example, if the question does not ask for data validation, then no marks will be awarded for data validation.
5. Your programs must be coded in such a way that they will work with any data and not just the sample data supplied or any data extracts that appear in the question paper.
6. Routines such as search, sort and selection must be developed from first principles. You may NOT use the built-in features of a programming language for any of these routines.
7. You must save your work regularly.
8. The files that you need to complete this question paper have been given to you on a disk/network. The files are given in the form of password-protected executable files:

Do the following:

- Double click on the password-protected executable file.
- Click on the extract button.
- Enter the following password: **School%4*#**

QUESTION 1

Question1_p.dpr
Question1_u.pas
Question1_u.dfm

QUESTION 2

clsLearners.pas
Question2_p.dpr
Question2_u.pas
Question2_u.dfm

QUESTION 3

examnumbers.txt
Question3_p.dpr
Question3_u.pas
Question3_u.dfm

SECTION A

QUESTION 1: GENERAL PROGRAMMING SKILLS

All Grade Twelve learners have to check their examination entry for the National Exam at the end of each year. Subject changes are allowed, but all supporting documents must be submitted. All information that is entered, must be correct.

INSTRUCTIONS:

- The project **Question1** is provided in the **TrialExam2016** folder.
- Open the incomplete project file called **Question1_p**.

1.1 Question 1.1 Button

The learner must enter his/her name and surname in the given text area. The relevant grade must be selected. The old and new subject must also be selected. If the old and new subject is the same, display an 'Incorrect Information' message. If no subject change is made, display the learner's details and 'No subject change'. If a subject change has been made, display the learner's information as well as a message indicating that all required documents should be submitted.

Example of output:

INCORRECT INFORMATION:

The screenshot shows a form titled 'Question 1'. It has a text field for 'Name of Learner' with 'Samantha Davies' entered. Below it are two radio buttons for 'Grade': '11' and '12', with '12' selected. To the right of the radio buttons are two dropdown menus, both set to 'Life Sciences'. A button labeled 'Question 1.1' is below the dropdowns. At the bottom, a text area displays 'Incorrect information'.

NO SUBJECT CHANGE:

The screenshot shows the same form as above. The 'Subject FROM' dropdown is set to 'Life Sciences' and the 'Subject TO' dropdown is also set to 'Life Sciences'. The 'Question 1.1' button is present. The text area at the bottom displays 'Learner: S Davies', 'Grade: 11', and 'NO SUBJECT CHANGE'.

SUBJECT CHANGE:

The screenshot shows the form with 'Name of Learner' as 'Samantha Davies', 'Grade' as '12' (selected), 'Mathematics' in the 'Subject FROM' dropdown, and 'Mathematical Literacy' in the 'Subject TO' dropdown. The 'Question 1.1' button is present. The text area at the bottom displays 'Learner: S Davies', 'Grade: 12', and 'Subject Change from Mathematics to Mathematical Literacy'. A dialog box titled 'Question1_p' is open over the form, with the message 'Please submit all the required documents' and an 'OK' button.

(13)

1.2 Question 1.2 Button

The ID number must be validated by comparing it with the date of birth. Display a suitable message to indicate whether it is correct or incorrect.

Example of output:

(8)

1.3 Question 1.3 Button

A username needs to be created. It is generated by removing all the vowels and spaces from the candidate's name and surname (as entered in Question 1.1); '%'; a random number between 100 and 300 (inclusive) and '#'. Example of output (Take note: The output will be different due to random numbers):

(11)

1.4 Question 1.4 Button

Code must be written to calculate whether the candidate has achieved a pass based on the marks they have entered.

Requirements for a pass are as follows:

- Home Language must be 40% or more
- 40% in three other subjects
- 30% in two subjects

Display on the panel whether the candidate has passed or failed.

Example of output:

1.5

HOME LANGUAGE	40
FIRST ADDITIONAL	56
LIFE ORIENTATION	65
MATHS OR MATHS LIT	24
Physical Sciences	26
Life Sciences	87
Information Technology	78

FAIL

Question 1.4

(15)

- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

TOTAL SECTION A: 47

SECTION B

QUESTION 2: OBJECT-ORIENTED PROGRAMMING

To get admission to tertiary institutions one must achieve a certain amount of Admission Points depending on the course and institution.

INSTRUCTIONS:

- The project **Question2** is provided in the **TrialExam2016** folder, which contains:
 - A main form unit called **Question2_u.pas**
 - An incomplete unit file called **clsLearners.pas**
- Open the incomplete project file called **Question2_p** and the class called **clsLearners.pas**

2.1 2.1.1 Declare an array with class scope which will store the candidate's marks (maximum 10 subjects). The private attributes are given; declare the public methods as you answer the questions below. (7)

2.1.2 Write code for the constructor method which will receive and assign the candidate's **name** and **number** of subjects. The attribute **fapspoints** must be initialised. (3)

2.1.3 Write a method called **APS** which will receive the marks from the array, convert it to the respective level and then calculate the APS points. APS points are calculated by converting the percentages to the respective levels and adding the levels together.

Percentage	Level
80-100	7
70-79	6
60-69	5
50-59	4
40-49	3
30-39	2
0-29	1

(4)

2.1.4 Write a method called **BachPass** which will receive the marks and return whether the learner has achieved a bachelor's pass.

In order to qualify for a bachelor pass, the learner must achieve at least 50% in four subjects. Assume that all the subjects are from the designated list of subjects.

(8)

2.1.5 Write a method called **CompileString** which will return the information as shown in the following format:

Learner's Name: <fname>
 Bachelor Pass: <Yes/No>
 APS Points: <fapspoints>

(3)

- 2.2 The learner's name must be entered in the text area; then the number of subjects that the learner is enrolled for must be entered using an InputBox component.

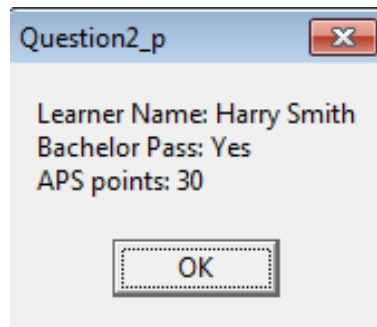
If the learner entered less than 7 subjects, then the learner must enter the number of subjects again until it is 7 subjects or more.

Use the name and number of subjects to instantiate the object **objLearner**.

The learner must enter the marks and these marks must be stored in an array. Once the marks are entered, then the program must check whether the learner qualifies for a bachelor pass. If the learner has a bachelor pass, then APS must be calculated. All the information must be displayed as shown in the example below. All relevant methods must be called.

If the learner does not qualify for a bachelor pass, a suitable message must be displayed.

Example of output:



(17)

- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

TOTAL SECTION B: 42

SECTION C**QUESTION 3: PROBLEM SOLVING PROGRAMMING**

A record of all the exam numbers and centres must be kept accurately for statistical purposes. Sometimes new candidates are added and some are deregistered for various reasons. It is also important to know how many candidates each centre has.

INSTRUCTIONS:

- The project **Question3** is provided in the **TrialExam2016** folder.
- Open the incomplete project file called **Question3_p**.

You are required to design a program that must be able to do the following:

- Get the exam numbers from a text file named **examnumbers.txt**
- Sort and display the exam numbers
- Search for a specific centre
- Add a new exam number to the array and text file
- Delete an exam number

Read the following sections before attempting the solution:

- GUI and data supplied
- Program requirements
- Mark allocation

NOTE:


- You may NOT modify data supplied.
- The use of good programming techniques and modular design must be applied in the design and coding of your solution.

GUI AND DATA SUPPLIED

GUI

The GUI contains components for input and output. Additional components will be required. At least one dynamic component must be used.

Example of GUI provided:



DATA

The exam centre number forms part of the candidate's exam number.

If the candidate's exam number is 4151610060031, then it can be interpreted as follows:

- The last four numbers (0031) represents the candidate's number at the centre.
- The centre number consists of the numbers following the positions: 1; 4 – 9 (4161006)
- The second and third number (15) can be ignored in this question.

The exam centres are stored in a one-dimensional array called **arrcentres**.

The examnumbers of the candidates are stored in a textfile named **examnumbers.txt**.

PROGRAM REQUIREMENTS

Get Information, Sort and Display

- Get the contents of the **examnumbers.txt** text file.
- Sort it in ascending order.
- Display the sorted exam numbers.

Search for centre number

- The user must enter a centre number.
- If the centre number is found, the school name as well as all the exam numbers from that particular centre must be displayed.
- The total number of candidates must be displayed as well.
- If the centre number is not found, a suitable message must be displayed.

Add exam centre and exam number

- Get the information that must be added from the user
- Add the exam number to the array as well as to the text file
- Display a message that the exam number has been added to the text file
- Sort and display the updated array

Delete an exam number

Sometimes candidates are deregistered due to various reasons.

- Ask the user to enter the exam number to be removed
- Delete the exam number from the text file as well as from the array
- Display a message once the exam number is deleted
- Sort and display the updated array
- Write the updated array to the existing text file

MARK ALLOCATION

REQUIREMENTS	MAXIMUM MARKS
Modular Design, Dynamic Component, Indentation	5
Get Information, Sort and Display	18
Search	15
Add	10
Delete	13

- Enter your name as a comment in the first line of the program file.
- Save the program.
- A printout of the code may be required.

TOTAL SECTION C: 61
GRAND TOTAL: 150

