



Province of the
EASTERN CAPE
EDUCATION

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

SEPTEMBER 2017

INFORMATION TECHNOLOGY P1

MARKS: 150

TIME: 3 hours

This question paper consists of 9 pages.



INSTRUCTIONS AND INFORMATION

1. This question paper consists of THREE questions. Candidates must answer ALL three questions.
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. 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.
4. 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.
5. 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.
6. Save your work regularly.
7. The files that you need to complete this question paper have been given to you. The files are provided 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: **Med2017#**

Once extracted, the following list of files will be available in the folder **DataTrial2017**:

Question 1:

Question1P.dpr
Question1P.res
Question1U.dfm
Question1U.pas
Medical.jpg

Question 2:

clsRecords.pas
Question2P.dpr
Question2P.res
Question2U.dfm
Question2U.pas

Question 3:

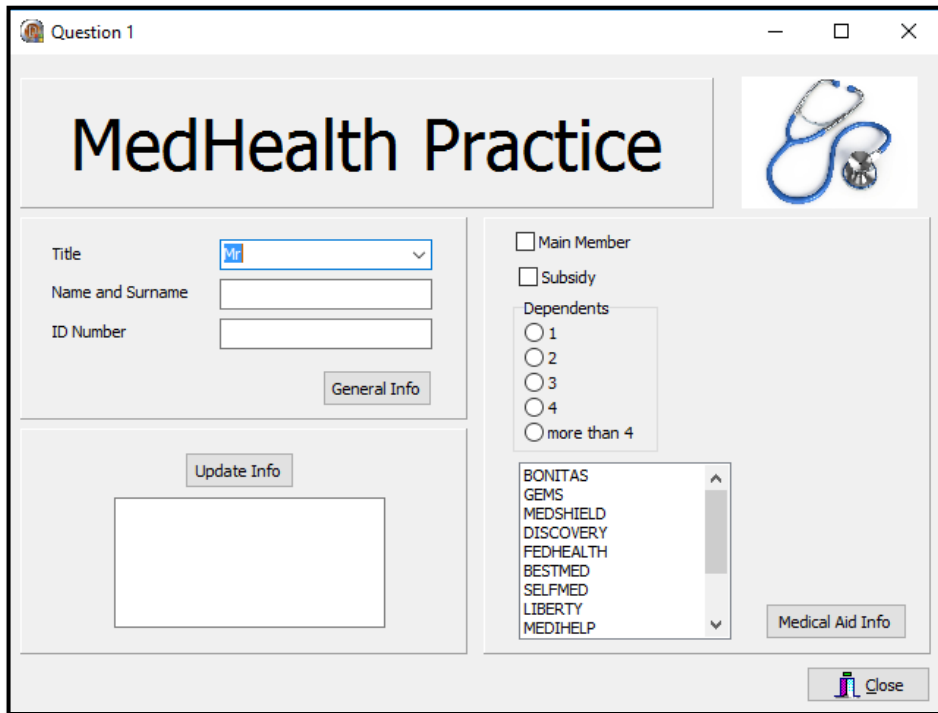
Question3P.dpr
Question3P.res
Question3U.dfm
Question3U.pas
Patients.txt

SCENARIO

Healthcare is a very important part of our lives. In this question paper we are going to complete programs mainly aimed at doctors and patients. Medical aids are also assumed to be a necessity.

QUESTION 1: GENERAL PROGRAMMING SKILLS

An example of the GUI is given below:



Complete the code for each section of QUESTION 1 as described in QUESTION 1.1 to QUESTION 1.4.

- 1.1 Add the picture, Medical.jpg, as a **dynamic component** when the program is executed.

Properties:

Left = 500
Top = 15
Height = 90
Width = 120
Stretch = True

(7)

- 1.2 **General Info Button:**

The patient must enter their title, name and surname, as well as their ID number. The ID number must be checked to ensure it is the correct length (13 characters) and that it consists only of numbers. If the ID number is not correct, display a suitable message.

If the ID number is correct, a file name will be created for the patient. The file name consists of the first three characters of the surname and a randomised number between 100 and 999 (both included).

Display a welcome message which will include the title, initial, surname and file name of the patient.

Example of input:

Example of output:

(14)

1.3 Medical Aid Info Button

The patient must enter their medical aid info:

- If the patient is the main member, (Main Member) must be displayed next to the file name
- If the patient has a subsidy then 'Yes' must be stored, else 'No'
- Number of dependents
- Medical aid name

This information must be written to text file. The text file's name must be the file name created in Question 1.2.

The information must be written to the text file as follows:

<file name>: (Main Member)
 <number of dependents>
 <medical aid name>
 <subsidy>

Display a message once the file has been written.

(13)

1.4 Update Info Button

The information must be updated annually. Display a message which will tell the patient when the information must be updated again. Make use of the system date.

Example of output:

(5)

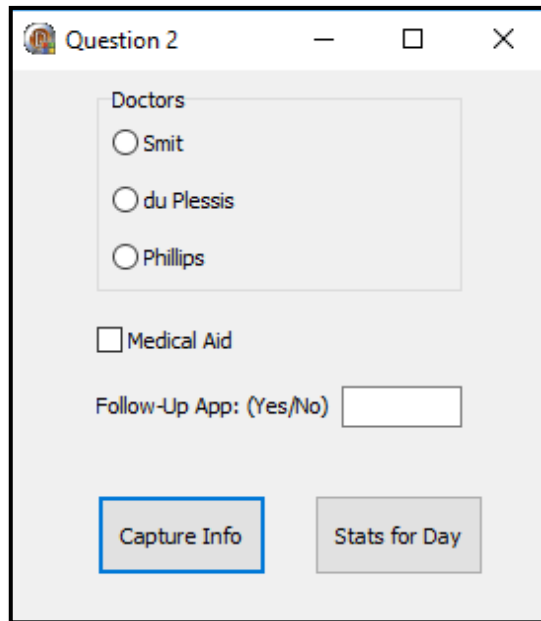
- Enter your name and surname as a comment in the first line of the program file.
- Save your program.
- A printout of the code may be required.

TOTAL QUESTION 1: [39]

QUESTION 2: OBJECT-ORIENTED PROGRAMMING

A receptionist has to manage many different tasks. Patients will want to make payments, book the next appointment and at the end of the day the total money received must also be recorded for financial purposes. The receptionist has to capture all the information for the day before she leaves the office.

Open the incomplete program in the Question 2 folder. Currently the program has no functionality.



Complete the code for his program as specified in QUESTION 2.1 and QUESTION 2.2.

2.1 The given incomplete object class contains the declaration of six attributes which describe the **objRecords** object.

| NAMES OF ATTRIBUTES | DESCRIPTION |
|---------------------|--|
| fdoctor | Name of the doctor |
| fdate | Date of doctor's visit using the system date |
| fmedaid | Yes/No indicating whether the patient has medical aid or not |
| ffollowup | Whether follow-up is needed ('Yes' or 'No') |
| fpayment | Cash payment made |
| fmed | Amount charged to medical aid |

2.1.1 Write a constructor called **Create** which will assign values to the fdoctor, fdate, fmedaid, ffollowup attributes. Fpayment and fmed must be initialised to zero. (4)

2.1.2 Write a method called **FollowUpdate** which will return the date the patient has to see the doctor again. The follow-up appointment has to be seven days after the first appointment. If no follow-up appointment is required then a string message must be returned indicating that no appointment is needed. (Assume that the month has 30 days.) (9)

- 2.1.3 Write a mutator method called **SetPayment** which will receive the payment that was made and update the existing payment amount. (2)
- 2.1.4 Write a mutator method called **SetMed** which will receive the amount that was submitted to the medical aid and update the existing medical aid amount. (2)
- 2.1.5 Write a method called **CompileString** which will return information about the follow-up appointment.

If a follow-up appointment is needed, the message should be the following:

Patient must please see <Doctor> on <follow-up date>

If no follow-up appointment is needed, then the following message should be displayed:

No Follow-Up Appointment Needed (5)

2.2 An incomplete class **Question2U** is provided.

2.2.1 **Capture Info Button**

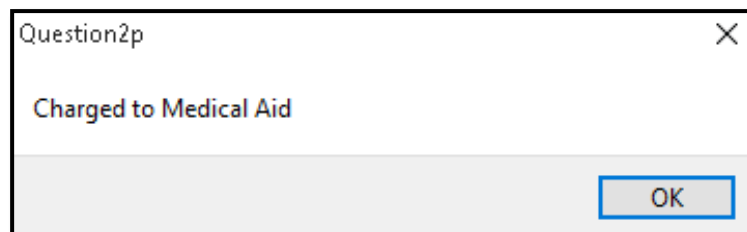
Write code to capture all the information and use it to instantiate the **objRecords** object. Call the **CompileString** method and display the contents using a message box.

The amount requested to be paid will be randomised and must be between 300 and 400 (both amounts included).

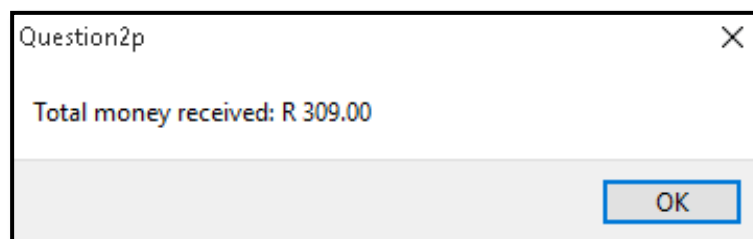
If the patient has a medical aid, then the amounts will be sent to the medical aid on behalf of the patient. If the patient has no medical aid, then the amount must be paid by the patient.

Example of output:

Patient has medical aid:



Patient does not have medical aid (take note that amounts will be different due to the amounts being randomised):



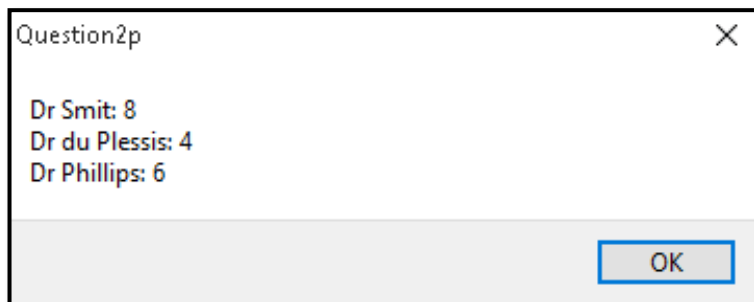
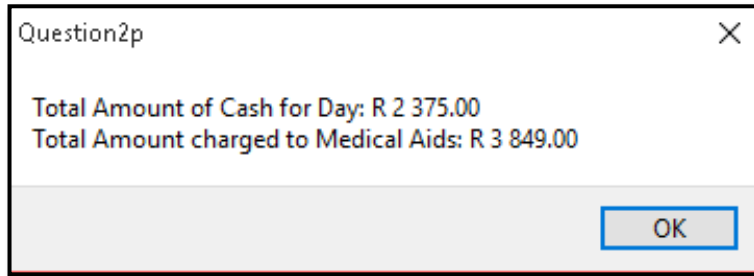
(20)

2.2.2 Stats for the Day Button

Write code to display the total amount of cash that was paid for the day as well as the total amount that was charged to medical aid. All monetary values must be formatted to two decimal places.

Display how many patients each doctor had on that particular day.

Example of output (output will be different as it depends how many times the Capture Info was selected; as well as randomising of numbers):



(5)

- 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 QUESTION 2: [47]

QUESTION 3: PROBLEM-SOLVING PROGRAMMING

This question will focus on the actual appointments. Two arrays have been given: **arrtimes** and **arrdoctors**.

3.1 When the program is executed, this should be displayed:

Write code to add the column (**arrtimes**) and row headings (**arrdoctors**).

The current patients' names are stored in a text file, **patients.txt**. Extract the names and store them in a two-dimensional array, with class scope, called **ar2appointments**.

The layout of the text file is as shown below.

```

patients.txt - Notepad
File Edit Format View Help
Dr du Plessis#Dr Smith#Dr Wessels#Dr Tom
-##John Dillon#-
-##Tanja Fourie#-
Jacky van Rensburg#-##James Marshall
-##Emily Smith#-#-
-##Conrad Williams#Leo Hendricks
-##Johan Middleton#Jessica Botha#-
-##Sam Smith#-
-##Hayden McAllistar#-
Azi Mbete#-##-#
Jack Turner#-##Sally Jacobs
  
```

Make use of a method to display the names in the string grid.

(21)

3.2.1 Insert Method

Write a method called **Insert** which will receive the name of the patient as well as the column (doctor's number) as parameters. A timeslot must be randomised. If the doctor's number is between 1 and 4 then the patient must be allocated to that specific doctor in a timeslot that is available. If the doctor number is 5, then the patient can be assigned to any doctor at random. The appointment must be added to the two-dimensional array.

Display a suitable message whether the appointment was made or not. (17)

3.2.2 Add Appointment Button

The patient must enter the preferred doctor's number. Use another Inputbox to enter the patient's name and surname. Call the **Insert** method and display the updated two-dimensional array in the string grid. (4)

3.3 Change Appointment Button

Enter the patient's name and surname. Find the patient's details in the two-dimensional array and then call the **Insert** method. The patient must see the same doctor as their original appointment. The original appointment must be deleted in the two-dimensional array. Display the updated two-dimensional array. (7)

3.4 Search Button

Search for the patient's name and surname that was entered. Display a message that will include the doctor who will be seen as well as the time of the appointment. (5)

3.5 Doctor Walk-In Button

The doctor with the least appointments on the day will be responsible for all the patients who need to see a doctor without a prior appointment. Write code to calculate which doctor has the least appointments and display a message indicating the doctor's name. (10)

- 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 QUESTION 3: [64]

TOTAL: 150