# GREENBURY SECONDARY SCHOOL

# INFORMATION TECHNOLOGY

## PAPER 1

## NOVEMBER EXAMS - 2018

## GRADE 11

MARKS: 150 TIME: 3 HOURS

This question paper consists of 13 pages including this cover page.

## INSTRUCTIONS AND INFORMATION

1. Answer **ALL** the 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. Make sure that you answer the questions according to the specifications that are given in each question. Marks will only be awarded based on the set requirements.

4. Only answer 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 it will work with any data and not just the sample data supplied or any data extracts that appear in the question paper.

6. **Make sure that you develop routines, such as search, sort and selection, from first principles and not use the built-in features of a programming language for any of these routines.**

7. Save your work regularly on the disk (CD/flash disk/DVD, et cetera) that you have been given, or on the disk space allocated to you for this examination.

8. Make sure that your **name and surname** appears as a comment in the first line of code. Also include the question number as part of the comment.

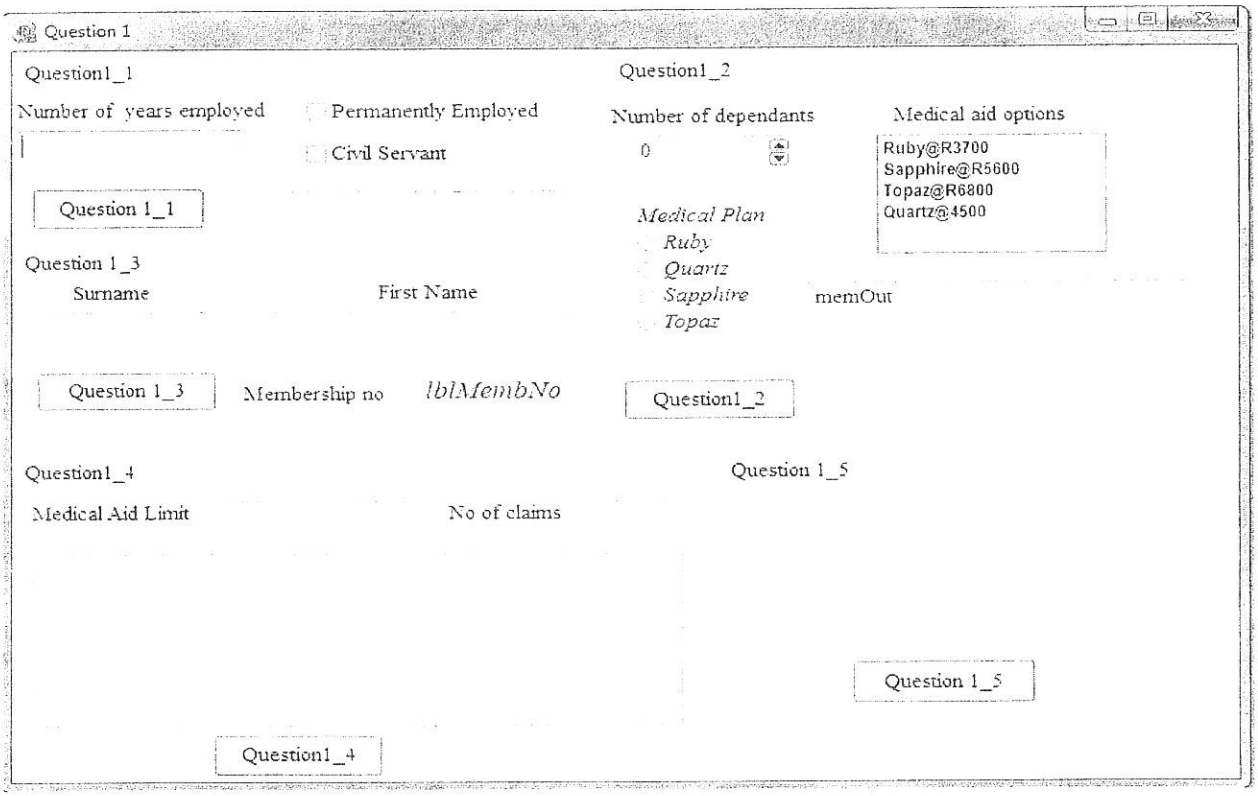
9. At the end of this examination session, you must hand in the disk/CD with all your work saved on it OR you must make sure that all your work has been saved on the disk space allocated to you. Ensure that all files can be read.

## SCENARIO

The primary focus in this paper is on healthcare. There are many aspects to Healthcare, which include hospitals, clinics, doctor surgeries and medical aid schemes. Your help is required to design/complete applications for some of these sectors

## QUESTION 1: *GENERAL PROGRAMMING SKILLS*

An excerpt of the GUI in project **Question1_U** is given below:




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

1.1 An individual is eligible to become a member of this medical aid scheme if

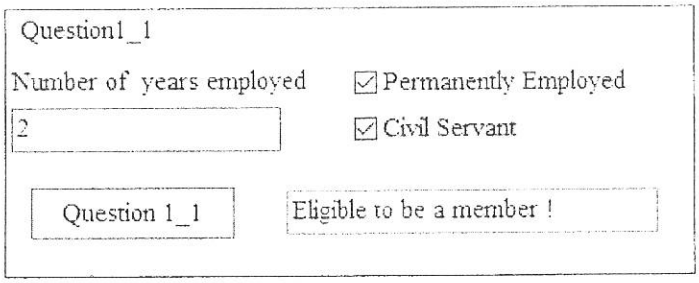
- ✓ they are permanently employed for more than 5 years

OR

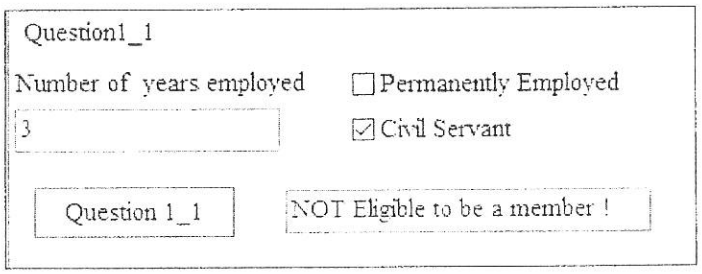
- ✓ any permanently employed civil servant (government employee ) can become a member *irrespective of the number of years employed*

**Write code to:**
retrieve the number of years employed from the text box , test whether the correct checkboxes are ticked and display a suitable message to indicate whether the person is eligible for membership to the medical aid scheme or not.

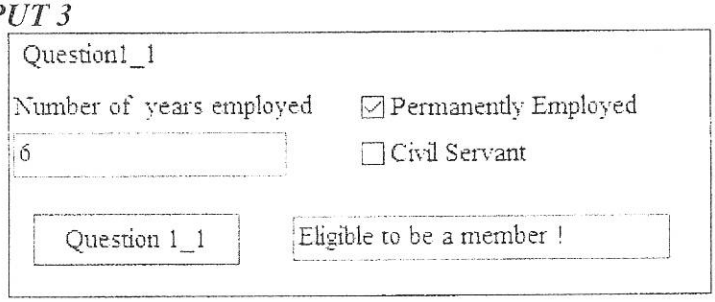
*SAMPLE OUTPUT 1*




*SAMPLE OUTPUT 2*



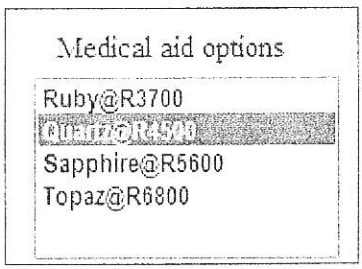

*SAMPLE OUTPUT 3*




(8)

1.2 A list box (lstMedPlan) is provided with the names of the different plans the medical scheme offers and their rates (tariff) per month for the main member.
The format of the text in the list box is as follows:

***<Medical plan name >@R<Tariff>***
***EXAMPLE***



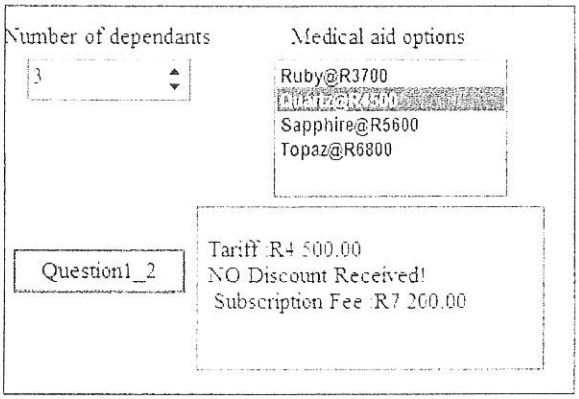

An amount of R900 is charged per dependant, which is added to the tariff amount, to obtain the members` subscription fee for the month.

**Write code to do the following**:

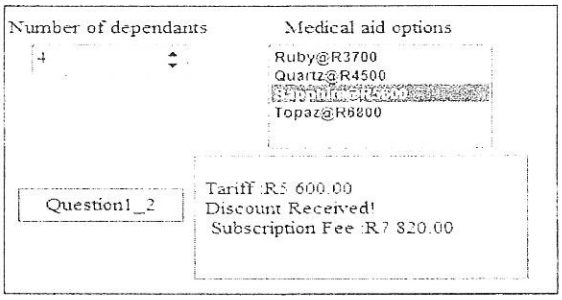
Read the number of dependants from the spin-edit *(spnDep)*, the tariff and the plan selected from the list box(*lstMedicalScheme*) then:

- Based on the selected option from the list box , write code to **select the corresponding option** for the medical aid plan in the group box (*rgpPlan*) Eg. If Quartz@R4500 is selected in the list box(*lstMedicalScheme*), then the Quartz option must be selected in the radio group(*rgpPlan*)
- Use the information obtained to calculate the subscription fees due.
- In addition, if the member has more than 3 dependants, than they qualify for a 15% discount off their subscription fees.
- Display a message whether the member qualifies for the discount or not, and the subscription fee due.

***SAMPLE OUTPUT 1***




***SAMPLE OUTPUT 2***




(10)

1.3 A person, approved for membership of the medical aid, needs to be assigned a membership number. The following strategy is used to generate a membership number for a member, after extracting the relevant information from the GUI components.

Membership number consists of:

- First three digits of Surname, in uppercase
- Random three digit ODD number, where the first digit should be a non-zero digit
- The symbol '~' (tilde) , used as a separator

- The letter C or X where C indicates that the member is a Civil Servant and X indicates they are not

Display the generated membership number in the label (*lblMembNo*).

**EXAMPLE OF MEMBERSHIP NO:** (12)
MAH539~C

1.4 Each medical aid plan has a medical aid limit. The following are the limits for each plan:

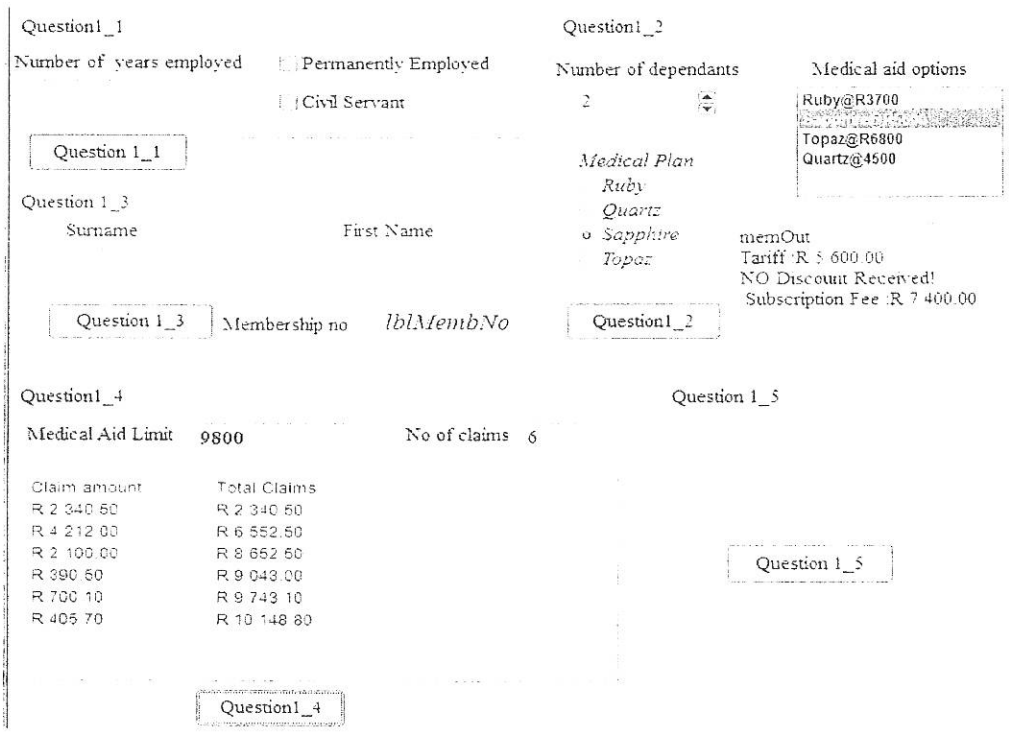
| Plan | Medical Limit |
|---|---|
| Ruby | R8000 |
| Sapphire | R9800 |
| Topaz | 2.5 times subscription fee |
| Quartz | R9000 |

- Use the medical aid option in 1.2 above to:
  determine the medical aid limit, based on the table above,
  and display the corresponding medical limit in the edit (*edtLimit*).

- Read in the claims made by the member in an input-box and display in tabular form the value of the claim made. and the total claims made. Use a conditional loop to continue reading the claim amounts, till the medical aid limit is reached, OR a value of -999 is entered for the claim amount.

- Keep a track of the total number of claims made and display this value in the edit (*edtClaims*). (10)

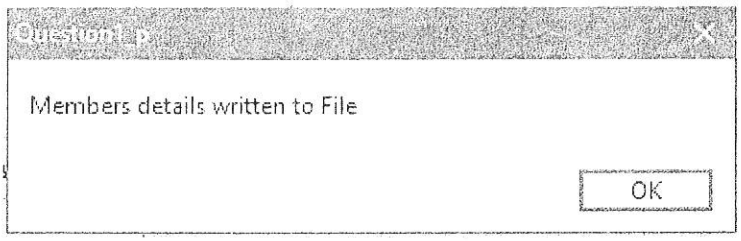
- Sample run:



1.5 A new text file, *members.txt,* containing member's details needs to be developed. Details of the member must be extracted from the GUI components and then written to the text file in the following format with captions:

**Member No** : <*Membership no*>
**Plan :** <*Medical Plan*>
**No of Dependants** : < *No of Dependants*>

Display a suitable message in a dialog box that members details have been written to File.




(10)

**TOTAL QUESTION 1: 50**

## QUESTION 2: ARRAYS & TEXT FILES

The *A.P. Memorial Hospital* keeps details of past patients who have outstanding debts with the hospital. The hospital would like to store the details of the debtors and the balances owing in two parallel arrays viz :

*arrDebtor*: should contain the name of the debtor (patient owing money)
*arrBalance*: should contain the amount being owed by the debtor

A textfile called *DataQ2.txt*, provided, contains details of the debtors, stored in the following format :

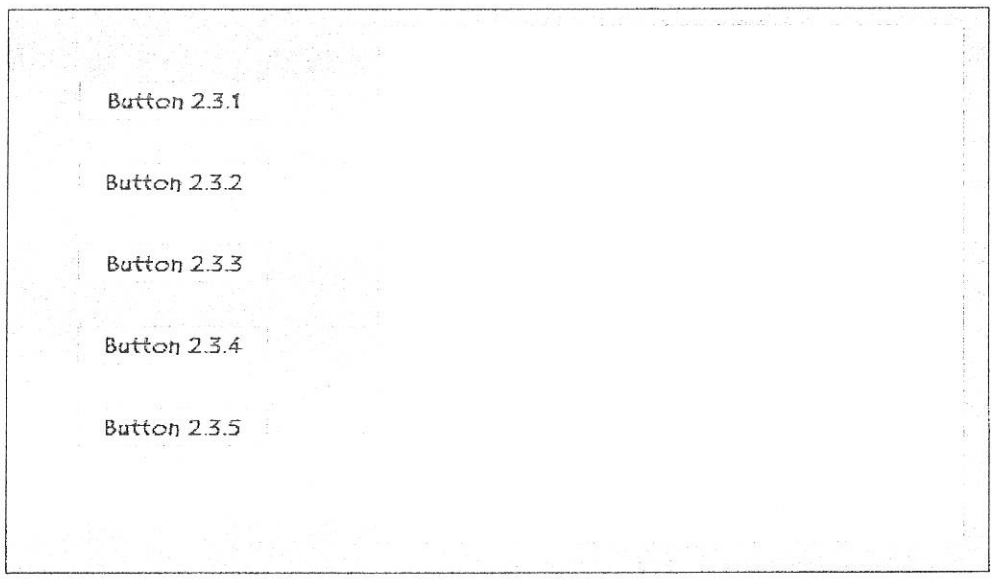
**Account No #Name of Debtor#Amount**

*Example*:
*A6795#Moodliar AT#4587.35*

***NOTE***::

- All output involving currency, should be displayed in currency format, where applicable.
- A global variable called, *count*, is declared to keep count of the number of entries in the array.

The form *Quest2_u* has the following interface:




2.1 Declare the above two arrays, to store the data from the text-file (2)

2.2 ***Code the following***:

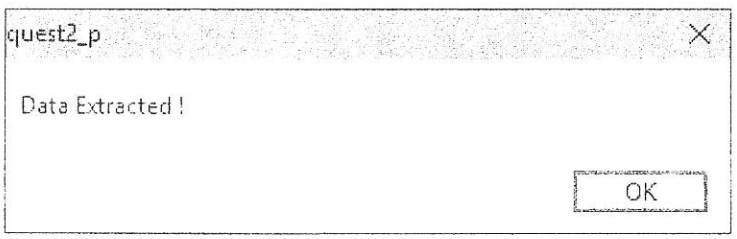
2.2.1 **Procedure Display:**
Write a procedure that would display the details of the debtors, together with their current balances in tabular form, with suitable headings. (5)

2.2.2 **Function Total:**
Which will calculate and return the total amount being owed by the debtors (5)

2.3 Create on-click events for the following buttons, as instructed:

2.3.1 **Button 2.3.1:**
Read the data from the text file and store the data into the respective arrays.
Follow the following process :

- Check if the text-file can be located then proceed
- If the text file cannot be found, display a suitable error message in a dialogue box and close the program
- Use the global variable to keep track of each entry stored in the arrays
- Proceed to read the text-file, a line at a time, then separate the details and store the data into the respective arrays
- Display a suitable message in a dialogue box, after the extraction of the data. (13)




*NOTE*: Only the name of the debtor and the amount needs to be stored.

2.3.2 **Button 2.3.2:**
Display details of the debtors, by calling the procedure written in *(2.1)* above

(2)

***SAMPLE OUTPUT***

| Debtor | Balance |
|---|---|
| Moodliar AT | R4 587.35 |
| Zungu BM | R6 354.21 |
| Carlson W | R7 561.30 |
| Van Wyk AK | R1 652.31 |
| Govender S | R2 347.24 |
| Singh A | R12 545.55 |
| Dlamini PD | R345.54 |
| Naidoo S | R5 250.21 |
| Pillay RE | R2 450.25 |
| Smith TW | R18 654.54 |
| Bantho TR | R3 544.34 |
| Mariah L | R8 755.33 |
| Chetty CL | R1 782.25 |
| Shange HP | R6 543.22 |
| Arumugam ST | R8 723.27 |
| Shaik MH | R5 422.32 |
| Buthelezi M | R9 391.32 |
| Mahommed SA | R2 124.26 |

2.3.3 **Button 2.3.3:**
Display the total fees owed by the debtors, then calculate and display the average fee being owed by debtors. Display output in the rich-edit with a suitable caption. Use the function declared in *(2.2)* in your code.

***SAMPLE OUTPUT***:

Total fees outstanding :R 108 034.80
Average balance R6 001.93

(4)

2.3.4 **Button 2.3.4:**
Display the details of the debtors, who owe less than R2500. Display output in the rich-edit in tabular form with a suitable caption. Also determine and output the number of debtors who fall in this category.

(5)

***SAMPLE OUTPUT***:

The following debtors have balance less than R2500:

| Debtor | Balance |
|---|---|
| Van Wyk AK | R1 652.31 |
| Govender S | R2 347.24 |
| Dlamini PD | R345.54 |
| Pillay RE | R2 450.25 |
| Chetty CL | R1 782.25 |
| Mahommed SA | R2 124.26 |

2.3.5 **Button 2.3.5 :**

All fees incur an annual debt fee increase of 5%. Adjust each of the debt amounts by this percentage amount.

Display a message in a dialog box, indicating that the fee increase has been effected! (4)


**TOTAL QUESTION 2: 40**

## QUESTION 3: DATABASE APPLICATION

The hospital has a database of current patients admitted to the hospital in a database called *HospitalDB.mdb.*

**The database has a table called *PatientsTB*, with the following field structure:**

| Field Name | Data Type |
|---|---|
| PatientID | Short Text |
| PatientName | Short Text |
| CdtnName | Short Text |
| Gender | Short Text |
| Age | Number |
| DateAdmitted | Date/Time |
| DaysStay | Number |

***SAMPLE DATA from PatientsTB***

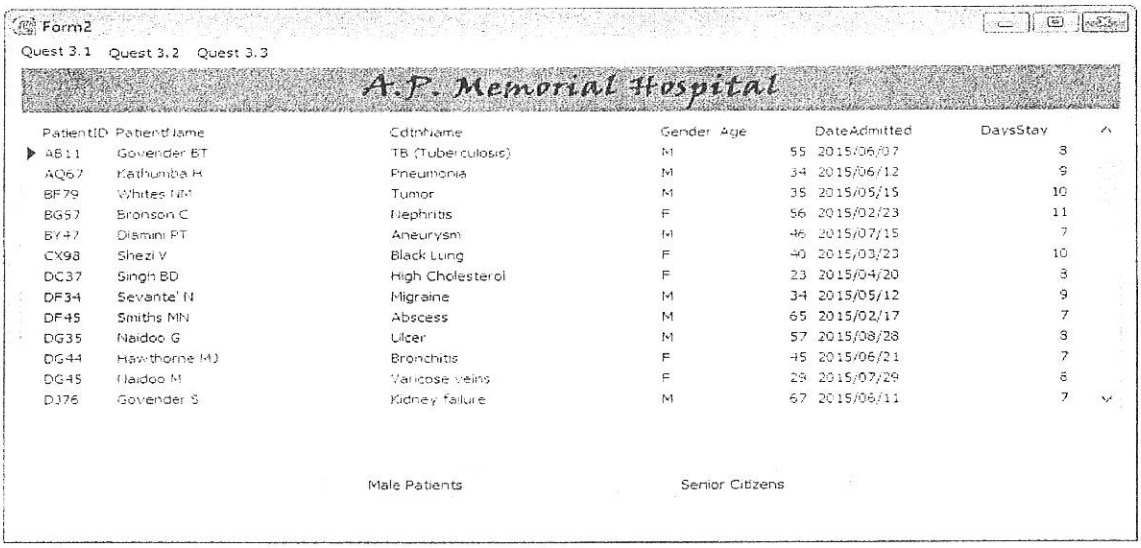
| PatientID | PatientName | CdtnName | Gender | Age | DateAdmitte | DaysStay |
|---|---|---|---|---|---|---|
| AB11 | Govender BT | TB (Tuberculosis) | M | 55 | 2015/06/07 | 8 |
| AQ67 | Kathumba H | Pneumonia | M | 34 | 2015/06/12 | 9 |
| BF79 | Whites NM | Tumor | M | 35 | 2015/05/15 | 10 |
| BG57 | Bronson C | Nephritis | F | 56 | 2015/02/23 | 11 |
| BY47 | Dlamini PT | Aneurysm | M | 46 | 2015/07/15 | 7 |
| CX98 | Shezi V | Black Lung | F | 40 | 2015/03/23 | 10 |
| DC37 | Singh BD | High Cholesterol | F | 23 | 2015/04/20 | 8 |
| DF34 | Sevante' N | Migraine | M | 34 | 2015/05/12 | 9 |
| DF45 | Smiths MN | Abscess | M | 65 | 2015/02/17 | 7 |
| DG35 | Naidoo G | Ulcer | M | 57 | 2015/08/28 | 8 |
| DG44 | Hawthorne MJ | Bronchitis | F | 45 | 2015/06/21 | 7 |

The program should be able to connect to the database named **HospitalDB.mdb**. If you find that the connectivity is not in place, use the following steps to establish connection with the database:

- ✓ Click on the TADOTable component .
- ✓ Click on the Ellipse button (three dots) to the right of the 'ConnectionString' property in the Object Inspector.
- ✓ Click on the Build button which takes you to the Data Link Properties dialogue box.
- ✓ The connection string is already defined. You may have to change the provider.
- ✓ Select Microsoft Jet 4.0 OLE DB if the database extension is .mdb.
- ✓ Click on Next.
- ✓ The first option on the Connection tabsheet allows you to browse and find the HospitalDB.mdb file.
- ✓ Remove the user name Admin.
- ✓ Click on the Test Connection button. Click OK on each one of the open dialogue windows.

NB. If you cannot establish connectivity with the database at all when you execute the program you must still code and submit the programming code for marking.

The following database application was developed to retrieve data from the database:



**Code on-click events for each of the following buttons as outlined below:**

3.1 Complete the code behind the following buttons in the *Quest 3.1* tab to display the required outcome in each case, using **CODE CONSRUCT/FILTERS AND NOT SQLs**.

3.1.1 **Male Patients**

Display the details of the male patients only. (3)

***SAMPLE OUTPUT***:

| PatientID | PatientName | CdtnName | Gender | Age | DateAdmitted | DaysStay |
|---|---|---|---|---|---|---|
| AB11 | Govender BT | TB (Tuberculosis) | M | 55 | 2015/06/07 | 8 |
| AQ67 | Kathumba H | Pneumonia | M | 34 | 2015/06/12 | 9 |
| BF79 | Whites NM | Tumor | M | 35 | 2015/05/15 | 10 |
| BY47 | Dlamini PT | Aneurysm | M | 46 | 2015/07/15 | 7 |
| DF34 | Sevante' N | Migraine | M | 34 | 2015/05/12 | 9 |
| DF45 | Smiths MN | Abscess | M | 65 | 2015/02/17 | 7 |
| DG35 | Naidoo G | Ulcer | M | 57 | 2015/08/28 | 8 |
| DJ76 | Govender S | Kidney failure | M | 67 | 2015/06/11 | 7 |
| DR55 | Zenade M | Stroke | M | 45 | 2015/07/18 | 8 |
| DT59 | Naidu P | Parkinson's disease | M | 56 | 2015/06/17 | 9 |
| FJ78 | Maharaj L | Alzheimers disease | M | 68 | 2015/03/16 | 7 |
| GB78 | Naicker T | TB (Tuberculosis) | M | 29 | 2015/02/19 | 8 |
| GB83 | Mahomed S | TB (Tuberculosis) | M | 37 | 2015/01/10 | 9 |

3.1.2 **Senior Citizens**

Display the details of all patients 60 years or older. (3)

***SAMPLE OUTPUT***:

| PatientID | PatientName | CdtnName | Gender | Age | DateAdmitted | DaysStay |
|---|---|---|---|---|---|---|
| DF45 | Smiths MN | Abscess | M | 65 | 2015/02/17 | 7 |
| DJ76 | Govender S | Kidney failure | M | 67 | 2015/06/11 | 7 |
| FJ78 | Maharaj L | Alzheimers disease | M | 68 | 2015/03/16 | 7 |

3.2 Complete the code behind the following buttons in the *Quest 3.2* tab, using appropriate **SQL** commands.

3.2.1 **Sort**

Write an SQL statement to display the name, gender and age of all patients, sorted in descending order of age (5)

***SAMPLE OUTPUT***:

| PatientName | gender | age |
|---|---|---|
| Maharaj L | M | 68 |
| Govender S | M | 67 |
| Smiths MN | M | 65 |
| Naidoo G | M | 57 |
| Dlamini KV | F | 57 |
| Hoosen ZK | F | 57 |
| Marimuthu M | F | 56 |
| Bronson C | F | 56 |
| Brown M | F | 56 |
| Doraivelu R | F | 56 |
| Naidu F | M | 56 |
| Govender BT | M | 55 |

3.2.2 **Females Over 50**

Write an SQL statement to display the patient ID, patient name and age of all female patients over 50 years of age. (6)

*SAMPLE OUTPUT*:

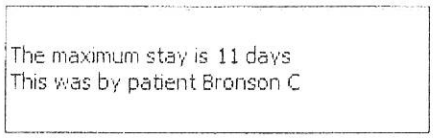
| patientID | PatientName | age |
|---|---|---|
| BG57 | Bronson C | 56 |
| DK10 | Doraivelu R | 56 |
| FG29 | Marimuthu M | 56 |
| FG47 | Ahmed AS | 54 |
| FH67 | Zungu MN | 53 |
| FH78 | Dlamini KV | 57 |
| FV56 | Brown M | 56 |
| GF76 | Hoosen ZK | 57 |

3.3 Complete the code behind the following buttons in the *Quest 3.3* tab, which would read through the data in the table and determine the required outcome and display in the output area(rich edit). **Use CODE CONSTRUCT AND NOT SQLs.**

3.3.1 **Longest Stay**

Write code to determine the name of the patient as well as the number of days in hospital, for the patient who has had the longest stay in the hospital.
Display output together with suitable statements, in the rich-edit. (7)

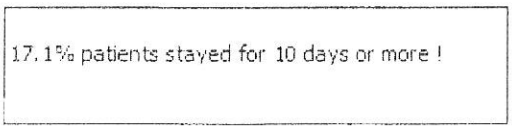
*SAMPLE OUTPUT*:




3.3.2 **Percentage**

Write code to calculate and display the percentage of patients whose stay in hospital was 10 days or more. Display output as a percentage correct to 1 decimal in the rich-edit, together with a suitable statement. (6)

***SAMPLE OUTPUT***:




**TOTAL QUESTION 3: 30**

**GRAND TOTAL = 120**