education

Department:
Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFORMATION TECHNOLOGY P1

NOVEMBER 2008

MARKS: 120

TIME: 3 hours

This question paper consists of 32 pages.

Copyright reserved

INSTRUCTIONS AND INFORMATION

1. This is a **three-hour** examination. 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.

2. Answer EITHER SECTION A (for Delphi programmers) OR SECTION B (for Java programmers).

3. You require the files listed below in order to answer the questions. They are EITHER on a stiffy disk OR CD issued to you OR the invigilator/teacher will tell you where to find them on the hard drive of the workstation you are using OR in a network folder:

   Question 1

   Delphi:
   LitterDBaseP.dpr
   LitterDBaseU.dfm
   LitterDBaseU.pas
   LitterCompDB.mdb
   Schools.txt
   Litter.txt

   Java:
   testLitter.java
   LitterCompDB.mdb
   Litter.java
   Schools.txt
   Litter.txt

   Question 2

   Delphi:
   testSchool_P.dpr
   testSchool_U.dfm
   testSchool_U.pas
   LitterWeight.txt

   Java:
   testSchoolLitter.java
   LitterWeight.txt

   Question 3:

   Delphi:
   Refuse_U.pas
   Refuse_U.dfm
   Refuse_P.dpr

   Java:
   testRefuseCollection.java
   RefuseCollection.java

   If a disk containing the above files was issued to you, write your name and examination number on the label.

4. Save your work at regular intervals as a precaution against power failures.

5. Save all your solutions in folders with the number of the question and your examination number as the name of the folder. For example: Quest2_3020160012.

6. Type in your examination number as a comment in the first line of each program.

7. Read ALL the questions carefully. Do not do more than is required by the question.

8. At the end of this examination session you will be required to hand in only the disk given to you by the invigilator or you must make sure that all the files have been saved on the network as explained to you by the invigilator/teacher. Please ensure that all files can be read.

9. During the examination you may make use of the manuals originally supplied with the hardware and software. You may also use the HELP functions of the software. ***Java candidates may make use of the Java API files*****.** *You may not refer to any other resource material.*

10. All printing of programming questions will take place within an hour of the completion of the examination.

Scenario:

As a project to help increase awareness of environmental issues the Department of Education has launched a ***Country Cleanup*** competition for schools. In each province, schools are allocated a specific area which they have to keep litter free.

Each week the litter that the school has collected is sorted into four categories, namely:

- Metal
- Glass
- Paper
- Other

The litter in each category is weighed and recorded at the end of every week.

SECTION A

Answer ALL the questions in this section only if you studied Delphi.

QUESTION 1: DELPHI PROGRAMMING AND DATABASE

The information on the schools participating in the litter competition during the first two weeks has been captured in a database named **LitterCompDB.mdb**. The organisers of the litter competition want you to write a program to make it easier for the administrative staff to answer certain queries from the public as well as participating schools.

The database, **LitterCompDB.mdb**, which contains two tables named **SchoolsTb** and **LitterTb** respectively, has been supplied to you in a folder named **Question 1 Delphi**.

If you cannot use the database provided, do the following:

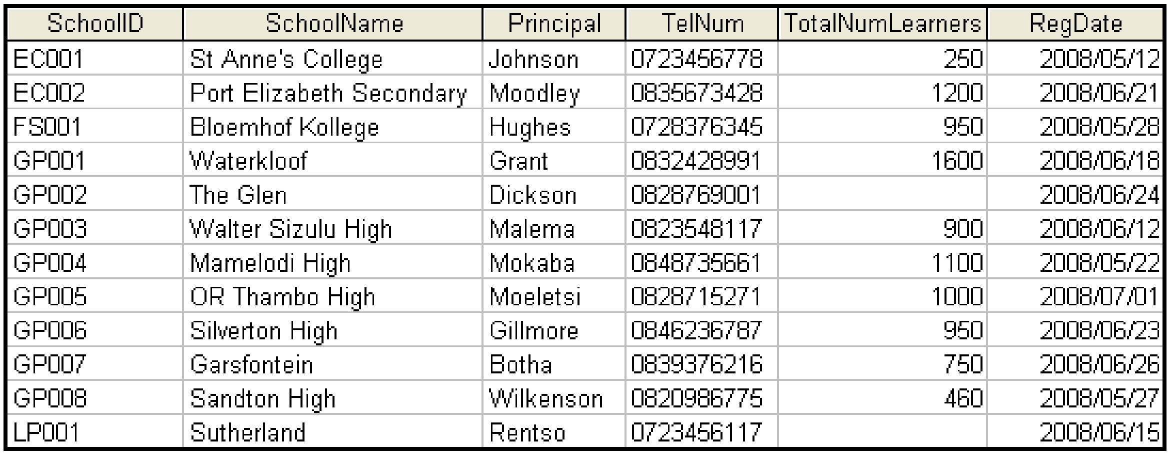
- Use the two text files named **Schools.txt** and **Litter.txt** supplied. Create your own database with a table named **SchoolsTb** and another table named **LitterTb** in the folder named **Question 1 Delphi**.

- Change the data types and the sizes of the fields in the two tables to the specifications given below (see next page). Create a one-to-many relationship (with referential integrity) between the **SchoolsTb** table and the **LitterTb** table.

The **SchoolsTb** table stores data on schools that are participating in the cleanup competition. The fields in this table are defined as follows:

Field Name	**Type**	**Size**	**Comment**
SchoolID	Text	5	Unique code
SchoolName	Text	25	
Principal	Text	20	Last name of the principal
TelNum	Text	15	Contact number
TotalNumLearners	Number	Integer	Total number of learners
RegDate	Date/Time	Shortdate	Date registered for competition

The following table is an example of the data contained in the **SchoolsTb** table.



SchoolID	SchoolName	Principal	TelNum	TotalNumLearners	RegDate
EC001	St Anne's College	Johnson	0723456778	250	2008/05/12
EC002	Port Elizabeth Secondary	Moodley	0835673428	1200	2008/06/21
FS001	Bloemhof Kollege	Hughes	0728376345	950	2008/05/28
GP001	Waterkloof	Grant	0832428991	1600	2008/06/18
GP002	The Glen	Dickson	0828769001		2008/06/24
GP003	Walter Sizulu High	Malema	0823548117	900	2008/06/12
GP004	Mamelodi High	Mokaba	0848735661	1100	2008/05/22
GP005	OR Thambo High	Moeletsi	0828715271	1000	2008/07/01
GP006	Silverton High	Gillmore	0846236787	950	2008/06/23
GP007	Garsfontein	Botha	0839376216	750	2008/06/26
GP008	Sandton High	Wilkenson	0820986775	460	2008/05/27
LP001	Sutherland	Rentso	0723456117		2008/06/15

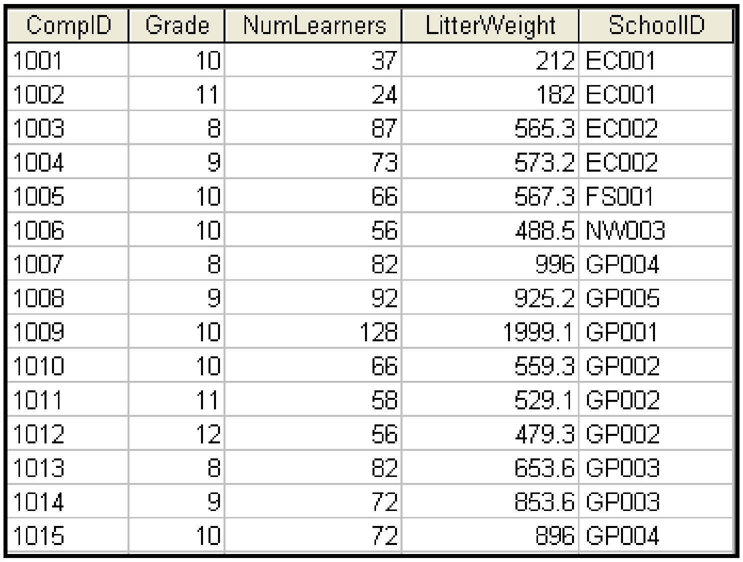
:

:

The fields in the **LitterTb** table are defined as follows:

Field Name	**Type**	**Size**	**Comment**
CompID	Text	4	Unique code of the school
Grade	Number	Byte	Grade of the group
NumLearners	Number	Integer	Number of learners in the group
LitterWeight	Number	Double	Total weight of litter in kilograms
SchoolID	Text	5	

The table (see the next page) is an example of the data contained in the **LitterTb** table.



CompID	Grade	NumLearners	LitterWeight	SchoolID
1001	10	37	212	EC001
1002	11	24	182	EC001
1003	8	87	565.3	EC002
1004	9	73	573.2	EC002
1005	10	66	567.3	FS001
1006	10	56	488.5	NW003
1007	8	82	996	GP004
1008	9	92	925.2	GP005
1009	10	128	1999.1	GP001
1010	10	66	559.3	GP002
1011	11	58	529.1	GP002
1012	12	56	479.3	GP002
1013	8	82	653.6	GP003
1014	9	72	853.6	GP003
1015	10	72	896	GP004

:

:

You have also been supplied with an incomplete Delphi program with a unit named **LitterDBaseU** and a project named **LitterDBaseP** in the folder named **Question 1 Delphi**. Open the incomplete program. It displays nine buttons as well as a DBGrid that will be used as an output component.

Do the following:

- Add your examination number to the caption of the form to the right of 'Question 1 –'.
- Change the **captions** of the buttons from top to bottom to match the screenshot (FIGURE 1.1) given below (see next page).

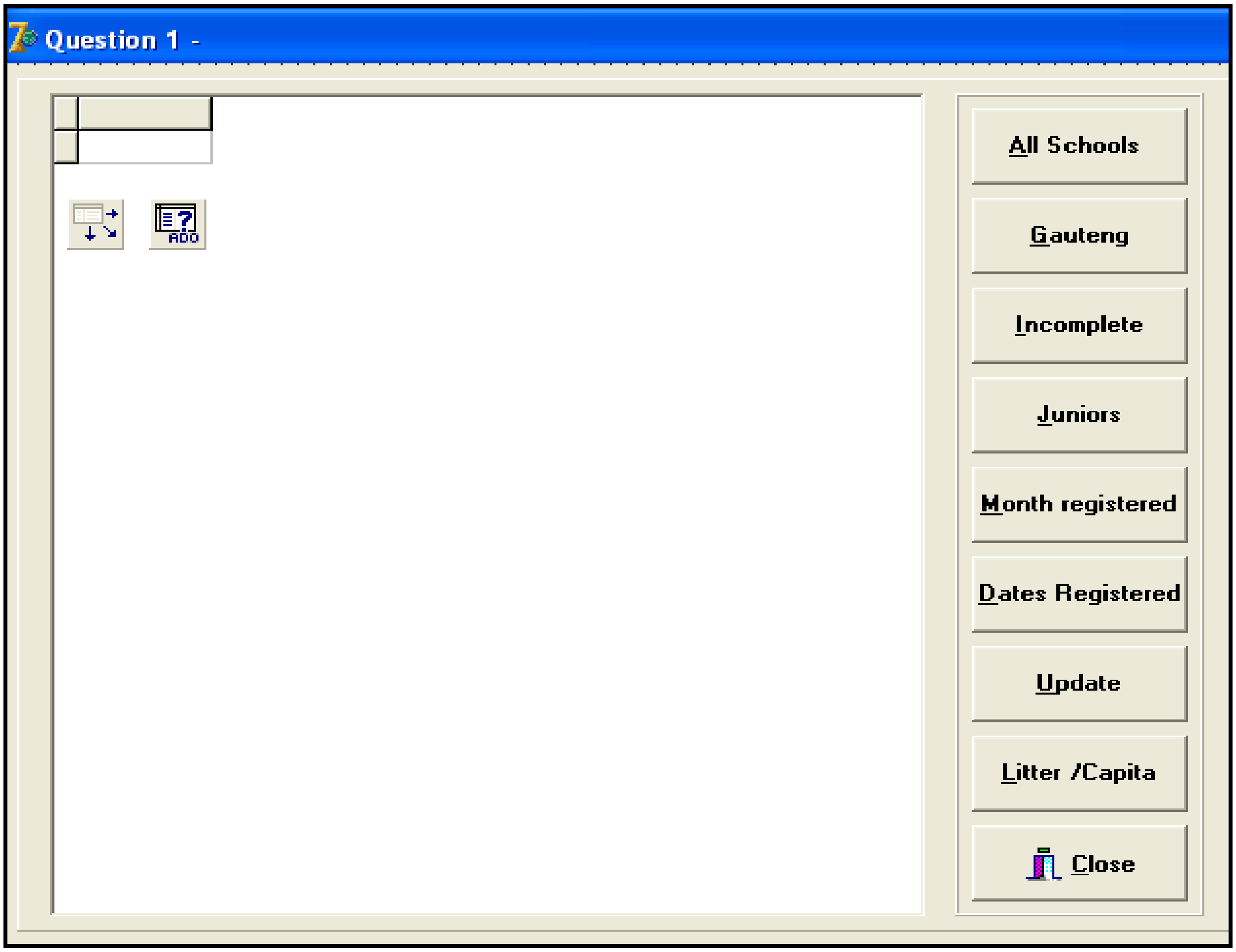


FIGURE 1.1

- Change the names of the buttons from top to bottom as follows: **btnAllSchools**, **btnGauteng**, **btnIncomplete**, **btnJuniors**, **btnMonthRegistered**, **btnDates**, **btnUpdateLearners** and **btnPerCapita**.

- The program should be able to connect to the database named **LitterCompDB.mdb**. When you do QUESTION 1.1 (which follows on the next page) and you find that the connectivity is not in place, use the following steps to establish connection with the database:

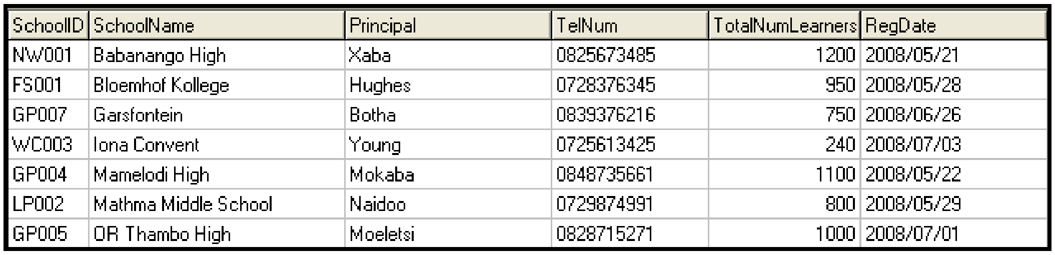
  - Click on the ADOQuery component named **qryLitter**
  - Click on the Ellipse button (three dots) to the right of the Connection string property in the Object Inspector
  - Click on the Build button which takes you to the Data Link Properties dialog box
  - Select Microsoft Jet 4.0 OLE DB Provider and click on Next
  - The first option on the Connection tab sheet allows you to browse and find the **LitterCompDB.mdb** file
  - Remove the user name Admin
  - Click on the Test Connection button
  - Click OK on each one of the open dialog windows

NOTE: If you cannot establish connectivity with the database when you execute the program you must still do the programming code and submit for marking.

Marks will only be awarded for the program methods that contain the SQL statements in the unit named LitterDBaseU.

Complete the program by creating the necessary SQL statements in the buttons named **btnAllSchools**, **btnGauteng**, **btnIncomplete, btnJuniors, btnMonthRegistered, btnDates, btnUpdateLearners** and **btnPerCapita** respectively as indicated in QUESTIONS 1.1 – 1.8.

1.1 Complete the code in the **All Schools** button by formulating a SQL statement to display all the fields in the **SchoolsTb** table in alphabetical order according to the names of the schools. Example of output:

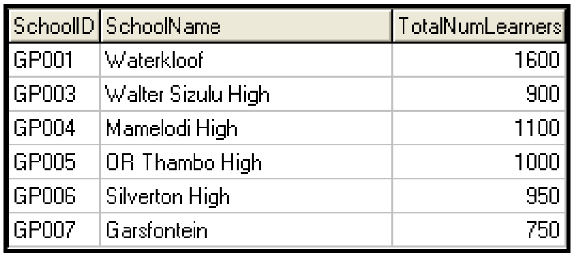


SchoolID	SchoolName	Principal	TelNum	TotalNumLearners	RegDate
NW001	Babanango High	Xaba	0825673485	1200	2008/05/21
FS001	Bloemhof Kollege	Hughes	0728376345	950	2008/05/28
GP007	Garsfontein	Botha	0839376216	750	2008/06/26
WC003	Iona Convent	Young	0725613425	240	2008/07/03
GP004	Mamelodi High	Mokaba	0848735661	1100	2008/05/22
LP002	Mathma Middle School	Naidoo	0729874991	800	2008/05/29
GP005	OR Thambo High	Moeletsi	0828715271	1000	2008/07/01

:

: (4)

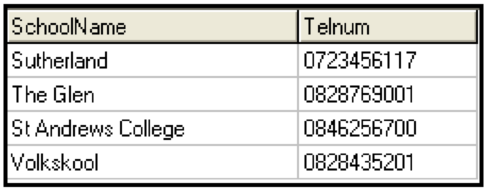
1.2 Complete the code in the **Gauteng** button by formulating an SQL statement to display the **SchoolID**, **SchoolName** and **TotalNumLearners** fields for the schools in Gauteng with more than 500 learners. The first two letters of the SchoolID indicates the province the school is from. GP indicates that the school is in Gauteng. Example of output:



SchoolID	SchoolName	TotalNumLearners
GP001	Waterkloof	1600
GP003	Walter Sizulu High	900
GP004	Mamelodi High	1100
GP005	OR Thambo High	1000
GP006	Silverton High	950
GP007	Garsfontein	750

(6)

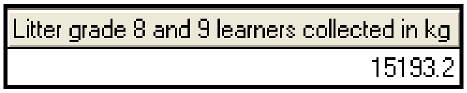
1.3 Complete the code in the **Incomplete** button by formulating an SQL statement to display the **Schoolname** and **TelNum** fields of the schools where the total number of learners in the school is not in the **SchoolsTb** table. Example of output:



SchoolName	Telnum
Sutherland	0723456117
The Glen	0828769001
St Andrews College	0846256700
Volkskool	0828435201

(3)

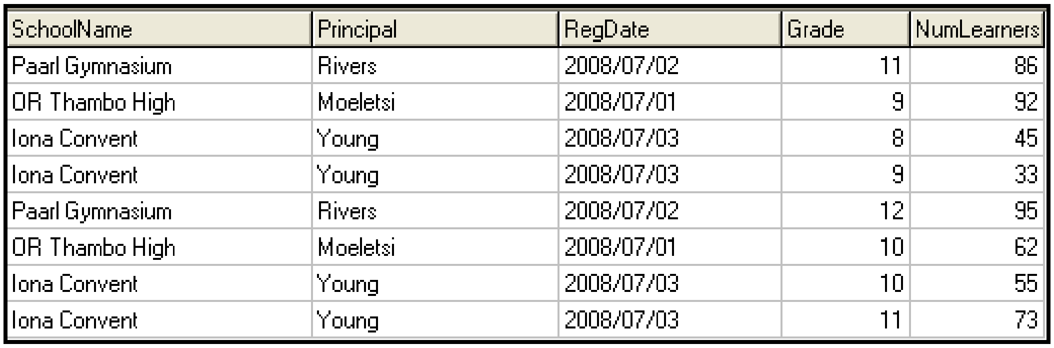
1.4 Complete the code in the **Juniors** button by formulating an SQL statement to determine the total weight of litter collected by the Grade 8 and 9 learners. Display the value as part of a user-friendly message. Example of the output:



Litter grade 8 and 9 learners collected in kg
15193.2

(5)

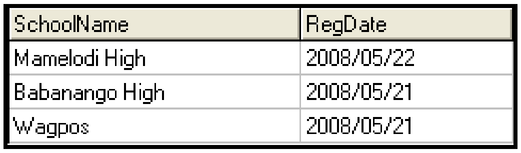
1.5 Complete the code in the **Month Registered** button by asking the user to enter the number of a month. Formulate an SQL statement to display the **Schoolname, Principal, Date Registered, Grade** and **Number of learners** of all the schools that registered during the month entered by the user. Example of the output where the user enters 7 as the number of the month:



SchoolName	Principal	RegDate	Grade	NumLearners
Paarl Gymnasium	Rivers	2008/07/02	11	86
OR Thambo High	Moeletsi	2008/07/01	9	92
Iona Convent	Young	2008/07/03	8	45
Iona Convent	Young	2008/07/03	9	33
Paarl Gymnasium	Rivers	2008/07/02	12	95
OR Thambo High	Moeletsi	2008/07/01	10	62
Iona Convent	Young	2008/07/03	10	55
Iona Convent	Young	2008/07/03	11	73

(6)

1.6 Complete the code in the **Dates Registered** button by formulating an SQL statement to list the **SchoolName** and **RegDate** fields of all the schools that registered on 21/05/2008 or 22/05/2008. Example of the output:

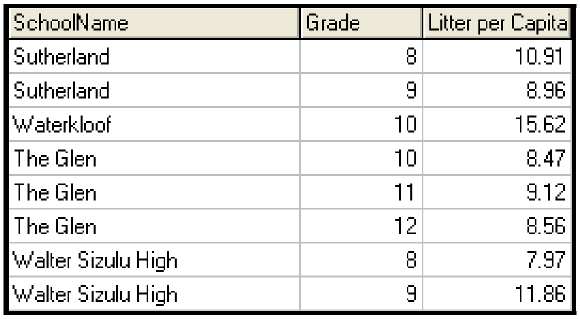


SchoolName	RegDate
Mamelodi High	2008/05/22
Babanango High	2008/05/21
Wagpos	2008/05/21

(5)

1.7 Complete the code in the **Update** button by asking the user to enter the name of the school and the number of the learners to be added in the **SchoolsTb**. Formulate an SQL statement to add the number of learners to the school and then display all the information of that school. (6)

1.8 Complete the code in the **Litter/Capita** button by formulating an SQL statement to list the school name and a calculated field (named **PerCapita**) to calculate and display the average litter collected per learner in each grade. Also display the name of the school and the grade. NOTE: You will need to link the tables with an appropriate ***where*** clause to be able to do this. Example of output:



SchoolName	Grade	Litter per Capita
Sutherland	8	10.91
Sutherland	9	8.96
Waterkloof	10	15.62
The Glen	10	8.47
The Glen	11	9.12
The Glen	12	8.56
Walter Sizulu High	8	7.97
Walter Sizulu High	9	11.86

:

: (5)

- Enter your examination number as a comment in the first line of the **LitterDBaseU** unit containing the SQL statements.
- Save the unit named **LitterDBaseU** and the project named **LitterDBaseP (File/Save All)**.
- Rename the folder **Question 1 Delphi** to **Quest1_X**, where X should be replaced with your examination number.
- A printout of the code for the **LitterDBaseU** unit will be required.

[40]

QUESTION 2: DELPHI – OBJECT-ORIENTED PROGRAMMING

This question is intended to test object-oriented programming skills. You are required to produce a solution that includes all classes specified in the instructions.

The amount of litter (in kilograms) collected by each school has been captured in a text file named **LitterWeight.txt** in the following format:

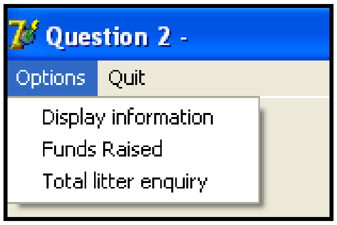
SchoolName#metal#glass#paper#other

An example of the data in the text file is:

**Sutherland#56.5#12.2#95.0#32.4
Waterkloof#32.4#67.9#19.1#43.2
The Glen High#103.2#35.6#9.3#0.5
Riverside High#46.7#72.2#53.6#39.4
Soweto High#82.3#89.9#29.1#13.2
Babanango High#123.3#33.3#49.3#4.5
Four Ways#77.7#13.2#89.9#132.2
Paarl Gym#65.1#87.9#44.1#73.2
Driehoek#3.2#135.6#67.3#55.5
OR Tambo High#156.5#32.2#25.2#34.3
Eden College#12.1#127.9#67.1#23.2
Iona Convent#73.5#85.1#99.3#15.5
Mamelodi High#66.6#62.2#96.0#62.6**

Do the following:

- Rename the folder **Question 2 Delphi** as **Quest2_X** (where X represents your examination number).
- Open Delphi and then open the file **testSchool_P.dpr** in the folder **Quest2_X**.
- Go to File/Save As … and save the unit as **testSchool_Uxxxx** (where xxxx represents the last FOUR digits of your examination number).
- Go to File/Save Project As … and save the project as **testSchool_Pxxxx** (where xxxx represents the last FOUR digits of your examination number).
- Change the caption properties of the different options of the Menu component so that it corresponds with the figure shown below.



- Add your examination number to the caption of the form to the right of 'Question 2 –'.

2.1 Create an object class (another unit) named **School_UXXXX** and save this unit as **School_UXXXX** in your **Quest2_X** folder. (XXXX should be replaced by the last four digits of your examination number.) Write the following code as part of this class:

2.1.1 Define a class named **TSchool**. This class must contain the following private fields:

fSchoolname
fMetalWeight
fGlassWeight
fPaperWeight
fOtherWeight

Ensure that you choose appropriate data types for these fields.
(4 ÷ 2) (2)

2.1.2 Write a parameterised constructor that will pass values for all the fields in the class. These parameters should be used to initialise the fields of the class. (4 ÷ 2) (2)

2.1.3 Write a method of type string (a string function) named **toString** that returns information on a school in one string formatted as follows:

Name of school, Metal weight, Glass weight, Paper weight, Other weight

Example of return strings for the first school in the text file **LitterWeight.txt**:

The Glen High 123.6 kg 59.5 kg 203.4 kg 303.3kg 689.8 kg
(8 ÷ 2) (4)

2.1.4 Write a method named **totalWeight** which calculates the total weight of all the categories of waste collected by a school and returns a decimal value. (4 ÷ 2) (2)

2.1.5 Write a method named **fundsRaised** that accepts four decimal values as parameters indicating the amount paid per kilogram for each category of waste. The method works out how much money the school has made by multiplying the weights of the appropriate category with the value of that category (e.g. fPaperWeight * fPaperValue). (6 ÷ 2) (3)

2.1.6 Write accessor (get) methods for all of the fields in the class.
(6 ÷ 2) (3)

2.2 Do the following in the **testSchool_Uxxxx** file (the main unit) in the given program:

2.2.1 Assign constant values to the four categories of weight in terms of rand per kilogram. Use the following values:

Metal waste:	R 1.50 per kg
Glass waste:	R 120 per kg
Paper waste:	R 1.00 per kg
Other waste:	R 0.50 per kg

(2 ÷ 2) (1)

2.2.2 Create an array named **arrSchools** that keeps objects of **TSchool**. Write code in the **OnActivate Eventhandler** of the form to read information from the text file **LitterWeight.txt** according to the following steps:

(a) Test if the text file exists. If it does exist, open the text file and initialise a loop to read the data. Display a suitable message if the file does not exist and terminate the program.

(b) Read a line of text from the text file.

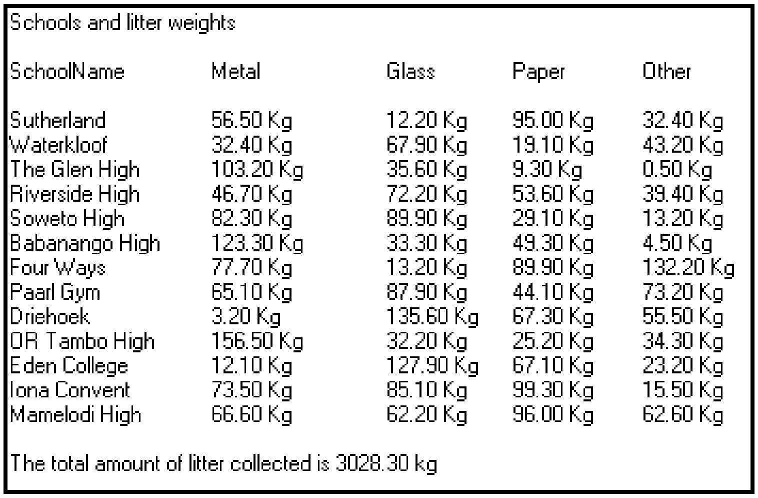
(c) Separate the text into the name of the school and the four weight figures indicating the kilograms of waste that have been collected by the school.

(d) Use this information to create a new **TSchool** object and place the object into the array.

(e) Use a counter field to keep track of how many items there are in the array. (24 ÷ 2) (12)

2.2.3 Write code to complete the options on the menu provided in the program. The methods in the **School_UXXXX** class should be used where applicable. Invoke the relevant methods (procedures and functions) from the class.

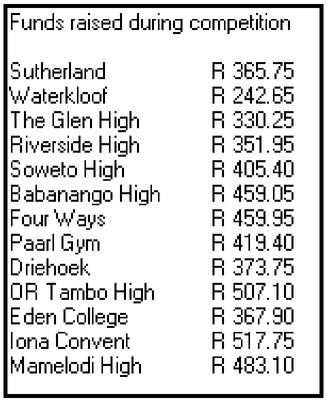
Display information: Display the names of the schools as well as the weights of the categories of waste they collected. Call the **toString** method to display the information. Display a suitable heading. Calculate and display the total amount of litter collected by the schools at the bottom of the list as indicated in the following example of the output:



```
Schools and litter weights

SchoolName        Metal           Glass       Paper       Other

Sutherland        56.50 Kg        12.20 Kg    95.00 Kg    32.40 Kg
Waterkloof        32.40 Kg        67.90 Kg    19.10 Kg    43.20 Kg
The Glen High     103.20 Kg       35.60 Kg    9.30 Kg     0.50 Kg
Riverside High    46.70 Kg        72.20 Kg    53.60 Kg    39.40 Kg
Soweto High       82.30 Kg        89.90 Kg    29.10 Kg    13.20 Kg
Babanango High    123.30 Kg       33.30 Kg    49.30 Kg    4.50 Kg
Four Ways         77.70 Kg        13.20 Kg    89.90 Kg    132.20 Kg
Paarl Gym         65.10 Kg        87.90 Kg    44.10 Kg    73.20 Kg
Driehoek          3.20 Kg         135.60 Kg   67.30 Kg    55.50 Kg
OR Tambo High     156.50 Kg       32.20 Kg    25.20 Kg    34.30 Kg
Eden College      12.10 Kg        127.90 Kg   67.10 Kg    23.20 Kg
Iona Convent      73.50 Kg        85.10 Kg    99.30 Kg    15.50 Kg
Mamelodi High     66.60 Kg        62.20 Kg    96.00 Kg    62.60 Kg

The total amount of litter collected is 3028.30 kg
```

(10 ÷ 2) (5)

Funds Raised: Display the detailed information on the funds raised by the schools during the waste collection competition. The program works out how much money the school has made by multiplying the weights of the appropriate category with the value of that category (e.g. fPaperWeight * fPaperValue). Display a suitable heading and subheadings. Example of the output:



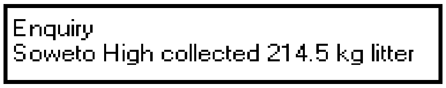
```
Funds raised during competition

Sutherland        R 365.75
Waterkloof        R 242.65
The Glen High     R 330.25
Riverside High    R 351.95
Soweto High       R 405.40
Babanango High    R 459.05
Four Ways         R 459.95
Paarl Gym         R 419.40
Driehoek          R 373.75
OR Tambo High     R 507.10
Eden College      R 367.90
Iona Convent      R 517.75
Mamelodi High     R 483.10
```

(8 ÷ 2) (4)

Total litter enquiry: Allow the user to enter the name of a school. Code an effective way to search for the name in the array by stopping the loop as soon as the name has been found.

Display the name of the school and the total weight of litter collected by the school. Display a suitable message if the name was not found. The name of the school must be part of the message.

Example of output when the user input is Soweto High:



```
Enquiry
Soweto High collected 214.5 kg litter
```

(14 ÷ 2) (7)

- Enter your examination number as a comment in the first line of the main unit (**testSchool_Uxxxx**) as well as the object unit (**School_UXXXX**).
- Save the files (File/Save All).
- Printouts of the code for the two units (**testSchool_Uxxxx** as well as **School_UXXXX**) will be required.

[45]

QUESTION 3: DELPHI PROGRAMMING

You are required to develop a program which summarises the results from the litter collection scheme run in 4 regions in a certain province for a number of schools. The number of schools participating in each region is not the same. The maximum number of schools that participated in a region was 6. The program should ask the user to enter the number of schools that participated in each region and then display a menu containing the options indicated in FIGURE 3.1 below.

The program should randomly generate data to populate the array.

You have been supplied with an incomplete program in the folder named **Question 3 Delphi**.

Do the following:

- Rename the folder named **Question 3 Delphi** to **Quest3_X**. Replace X with your examination number.
- Open the Delphi program in this folder.
- Save the unit as (File/Save As) as **Refuse_UXXXX** and the project as (File/Save project As) **Refuse_PXXXX** inside the folder (xxxx should be replaced by the last four digits of your examination number)
- Add your examination number to the caption of the form to the right of 'Question 3 –'.
- Change the captions to match those of the menu below (FIGURE 3.1):

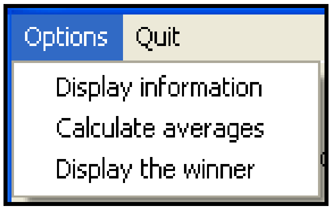
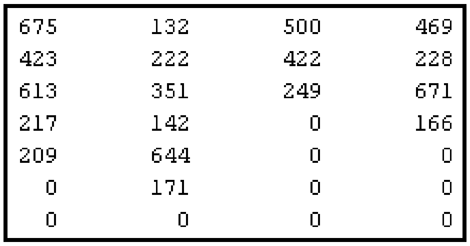


FIGURE 3.1

3.1 You are required to do the following:

3.1.1 Write code in the **OnActivate Eventhandler** to store the weight of litter collected by each school in a two-dimensional array which you must declare. The array will hold data for 4 regions and a maximum of 6 schools in each region. An extra row must be provided to store the total weight of litter collected by the schools in a region, for example:



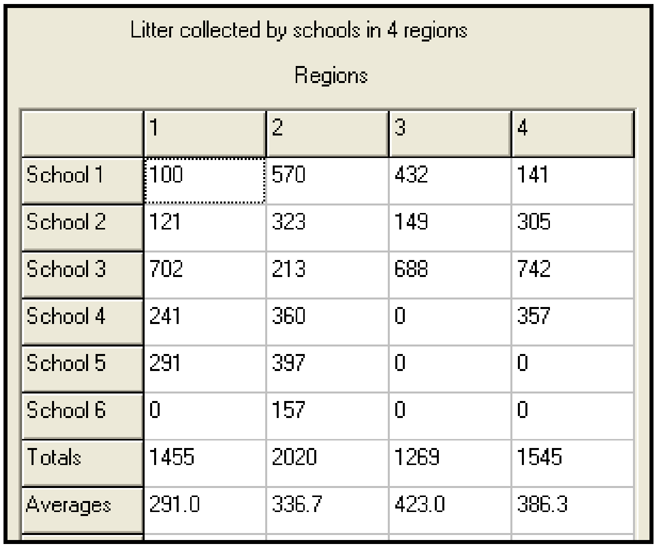
```
675        132        500        469
423        222        422        228
613        351        249        671
217        142          0        166
209        644          0          0
  0        171          0          0
  0          0          0          0
```

The array must be populated as follows:

- The program must request the user to input the number of schools in the first region which will be represented by the first column in the array. The weight of litter collected by each school is randomly generated and ranges from 100 to 800 kg and placed in the first column of the array.
- The same process will be repeated for the second, third and the fourth columns of the array. (18 ÷ 2) (9)

3.1.2 Write code in a subprogram to calculate the total and the average weights of litter collected by the schools in each region. Remember that each region may have a different number of schools. The average must be rounded off to one decimal place. (14 ÷ 2) (7)

3.1.3 Write code in a subprogram to display the contents of the two-dimensional array with a suitable heading and suitable labels. Also display averages at the bottom of the array. Example of output:



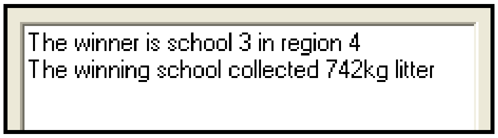
Litter collected by schools in 4 regions

Regions

	1	2	3	4
School 1	100	570	432	141
School 2	121	323	149	305
School 3	702	213	688	742
School 4	241	360	0	357
School 5	291	397	0	0
School 6	0	157	0	0
Totals	1455	2020	1269	1545
Averages	291.0	336.7	423.0	386.3

(20 ÷ 2) (10)

3.1.4 Write code in a subprogram to determine and display the school that collected the most litter in a RichEdit. Compile a message indicating the number of the school, the region where the school is situated and the weight of litter the school collected. Example of output:



```
The winner is school 3 in region 4
The winning school collected 742kg litter
```

(14 ÷ 2) (7)

3.2 When the program is executed the code in the OnActivate Eventhandler will be executed and then it should display a menu as shown in FIGURE 3.1 (see page 16).

For the option '**Display information**' you must ensure that the code to be executed will display the contents of the array as well as the averages of the regions.

NOTE: The totals and averages will have values once the 'Calculate averages' option has been executed.

NOTE: The values will be different every time you run the program.

For the option '**Calculate averages**' you must ensure that the code to be executed will calculate the totals and the averages.

NOTE: Click on the 'Display Information' to see the output.

For the option '**Display the winner**' you must ensure that the code to be executed will determine and display the information on the winning school.

(4 ÷ 2) (2)

- Enter your examination number as a comment in the first line of the unit **Refuse_UXXXX**.
- Save the unit and the project (File/Save All).
- A printout of the code for the unit **Refuse_UXXXX** will be required.

[35]

TOTAL SECTION A: 120

SECTION B

Answer ALL the questions in this section only if you studied Java.

QUESTION 1: JAVA PROGRAMMING AND DATABASE

The information on the schools participating in the litter competition during the first two weeks has been captured in a database named **LitterCompDB.mdb**. The organisers of the litter competition want you to write a program to make it easier for the administrative staff to answer certain queries from the public as well as participating schools.

The database, **LitterCompDB.mdb**, which contains two tables named **SchoolsTb** and **LitterTb** respectively, has been supplied to you in a folder named **Question 1 Java**.

If you cannot use the database provided, do the following:

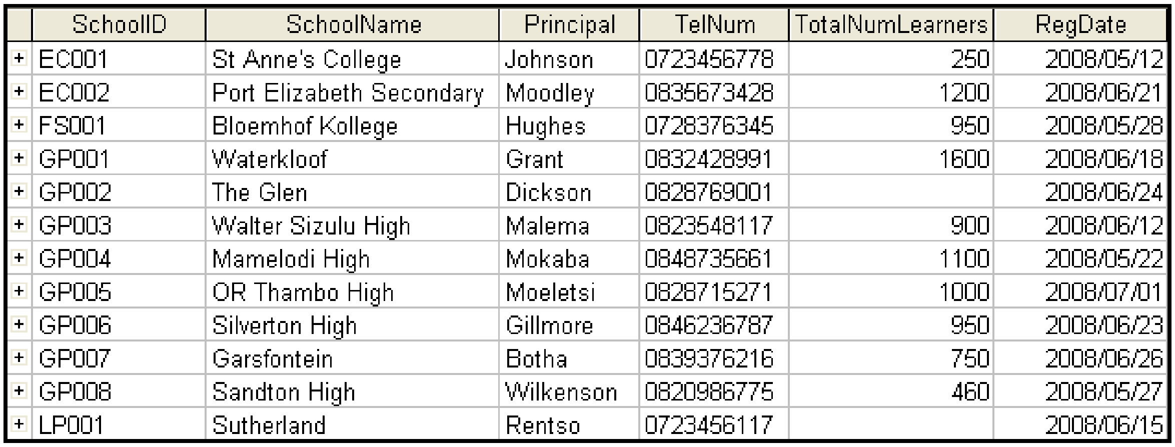
- Use the two text files named **Schools.txt** and **Litter.txt** supplied. Create your own database with a table named **SchoolsTb** and another table named **LitterTb** in the folder named **Question 1 Java**.

- Change the data types and the sizes of the fields in the two tables to the specifications given below. Create a one-to-many relationship (with referential integrity) between the **SchoolsTb** table and the **LitterTb** table.

The **SchoolsTb** table stores data on schools that are participating in the cleanup competition. The fields in this table are defined as follows:

Field Name	**Type**	**Size**	**Comment**
SchoolID	Text	5	Unique code
SchoolName	Text	25	
Principal	Text	20	Last name of the principal
TelNum	Text	15	Contact number
TotalNumLearners	Number	Integer	Total number of learners
RegDate	Date/Time	Shortdate	Date registered for competition

The following table is an example of the data contained in the **SchoolsTb** table.



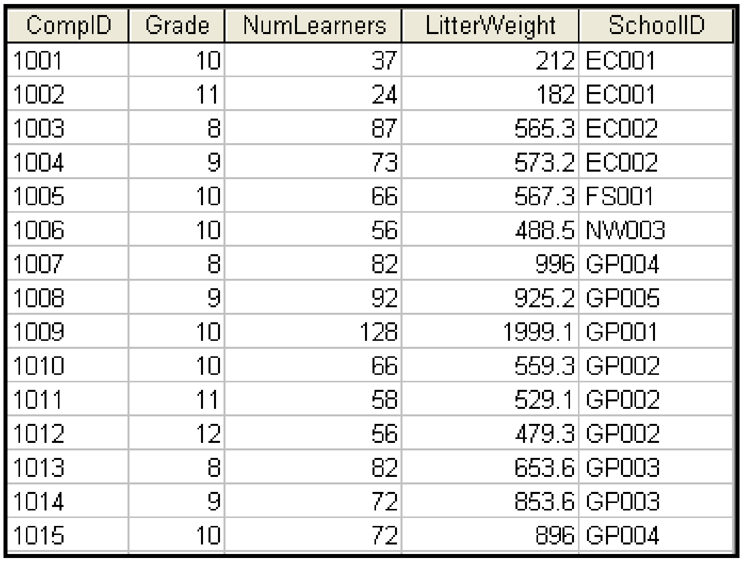
	SchoolID	SchoolName	Principal	TelNum	TotalNumLearners	RegDate
+	EC001	St Anne's College	Johnson	0723456778	250	2008/05/12
+	EC002	Port Elizabeth Secondary	Moodley	0835673428	1200	2008/06/21
+	FS001	Bloemhof Kollege	Hughes	0728376345	950	2008/05/28
+	GP001	Waterkloof	Grant	0832428991	1600	2008/06/18
+	GP002	The Glen	Dickson	0828769001		2008/06/24
+	GP003	Walter Sizulu High	Malema	0823548117	900	2008/06/12
+	GP004	Mamelodi High	Mokaba	0848735661	1100	2008/05/22
+	GP005	OR Thambo High	Moeletsi	0828715271	1000	2008/07/01
+	GP006	Silverton High	Gillmore	0846236787	950	2008/06/23
+	GP007	Garsfontein	Botha	0839376216	750	2008/06/26
+	GP008	Sandton High	Wilkenson	0820986775	460	2008/05/27
+	LP001	Sutherland	Rentso	0723456117		2008/06/15

:

The fields in the **LitterTb** table are defined as follows:

Field Name	**Type**	**Size**	**Comment**
CompID	Text	4	Unique code of the school
Grade	Number	Byte	Grade of the group
NumLearners	Number	Integer	Number of learners in the group
LitterWeight	Number	Double	Total weight of litter in kilograms
SchoolID	Text	5	

The following table is an example of the data contained in the **LitterTb** table.



CompID	Grade	NumLearners	LitterWeight	SchoolID
1001	10	37	212	EC001
1002	11	24	182	EC001
1003	8	87	565.3	EC002
1004	9	73	573.2	EC002
1005	10	66	567.3	FS001
1006	10	56	488.5	NW003
1007	8	82	996	GP004
1008	9	92	925.2	GP005
1009	10	128	1999.1	GP001
1010	10	66	559.3	GP002
1011	11	58	529.1	GP002
1012	12	56	479.3	GP002
1013	8	82	653.6	GP003
1014	9	72	853.6	GP003
1015	10	72	896	GP004

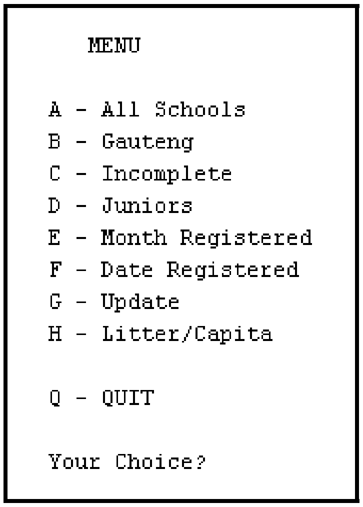
:

:

You have also been supplied with an incomplete Java program with a test class named **testLitter.java** and an object class named **Litter.java** in the folder named **Question 1 Java**. Open the incomplete program. The program will not run because of the incomplete SQL-statements. Once you have entered the correct SQL-statements, the program will display a simple menu with nine options.

Do the following:

- Change the code in the **testLitter** class to display the menu options (from top to bottom) to match the screenshot (FIGURE 1.1) given on the next page.



```
    MENU

A - All Schools
B - Gauteng
C - Incomplete
D - Juniors
E - Month Registered
F - Date Registered
G - Update
H - Litter/Capita

Q - QUIT

Your Choice?
```

FIGURE 1.1

- Change the names of the methods in the two Java files as follows:

  method A to selectAllSchools(),
  method B to selectGauteng(),
  method C to selectIncomplete(),
  method D to selectJuniors(),
  method E to selectMonth(),
  method F to selectDates(),
  method G to updateLearners(),
  method H to calcLitterPerCapita().

The connectivity code has already been written as part of the given code in the file named **Litter.java**. When you run the program, you have to enter the exact path where the database **LitterCompDB.mdb** has been stored.

HINT: Copy the database into the root directory of the drive that you are working on. The path to enter should be short, for example E:/LitterCompDB.mdb.

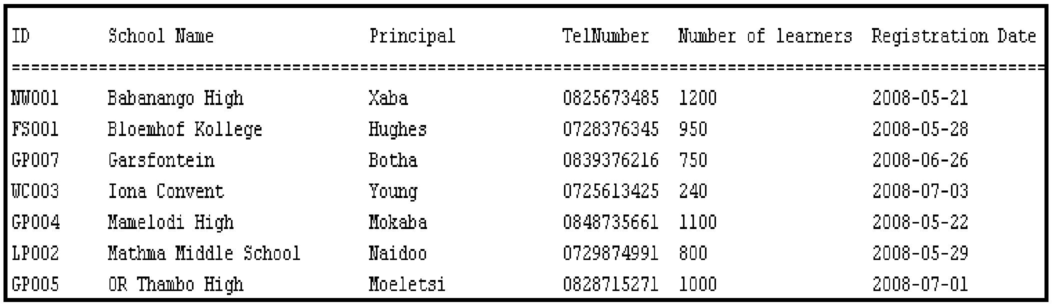
HINT: Instead of entering the path every time that you run the program, you can change the input to a constant string containing the exact location of the database, for example
String file name = 'E:/LitterCompDB.mdb'

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

Marks will only be awarded for the programming code which contains the SQL statements in the program named Litter.java.

- Complete the programming code in the file named **Litter.java** by creating the necessary SQL statements in the methods named **selectAllSchools(), selectGauteng()**, **selectIncomplete(), selectJuniors(), selectMonth(), selectDates(), updateLearners()** and **calcLitterPerCapita()** respectively as indicated in QUESTIONS 1.1 – 1.8.

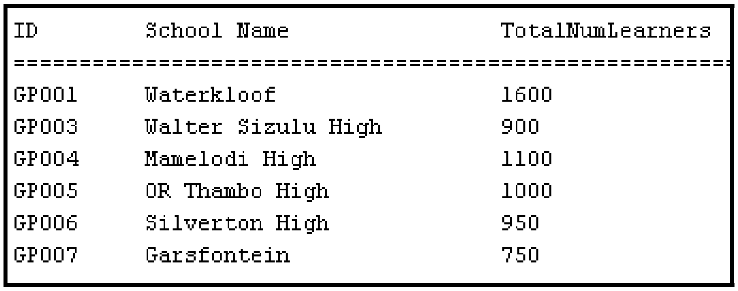
1.1 Complete the code in the **selectAllSchools** method by formulating an SQL statement to display all the fields in the **SchoolsTb** table in alphabetical order according to the names of the schools. Example of output:



```
ID       School Name             Principal        TelNumber   Number of learners  Registration Date
=====================================================================================================
NW001    Babanango High          Xaba             0825673485  1200                2008-05-21
FS001    Bloemhof Kollege        Hughes           0728376345  950                 2008-05-28
GP007    Garsfontein             Botha            0839376216  750                 2008-06-26
WC003    Iona Convent            Young            0725613425  240                 2008-07-03
GP004    Mamelodi High           Mokaba           0848735661  1100                2008-05-22
LP002    Mathma Middle School    Naidoo           0729874991  800                 2008-05-29
GP005    OR Thambo High          Moeletsi         0828715271  1000                2008-07-01
```

:
: (4)

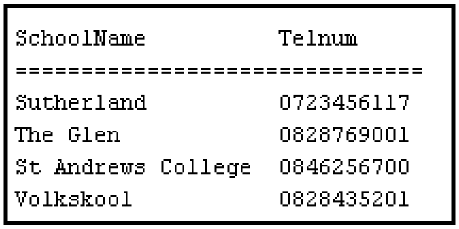
1.2 Complete the code in the **selectGauteng** method by formulating an SQL statement to display the **SchoolID**, **Schoolname** and **TotalNumLearners** fields for the schools in Gauteng with more than 500 learners. The first two letters of the SchoolID indicates the province the school is from. GP indicates that the school is in Gauteng. Example of output:



```
ID        School Name              TotalNumLearners
=====================================================
GP001     Waterkloof               1600
GP003     Walter Sizulu High       900
GP004     Mamelodi High            1100
GP005     OR Thambo High           1000
GP006     Silverton High           950
GP007     Garsfontein              750
```

(6)

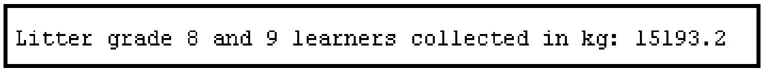
1.3 Complete the code in the **selectIncomplete** method by formulating an SQL statement to display the **Schoolname** and **TelNum** fields of the schools where the total number of learners in the school does not appear in the **SchoolsTb** table. Example of output:



```
SchoolName          Telnum
================================
Sutherland          0723456117
The Glen            0828769001
St Andrews College  0846256700
Volkskool           0828435201
```

(3)

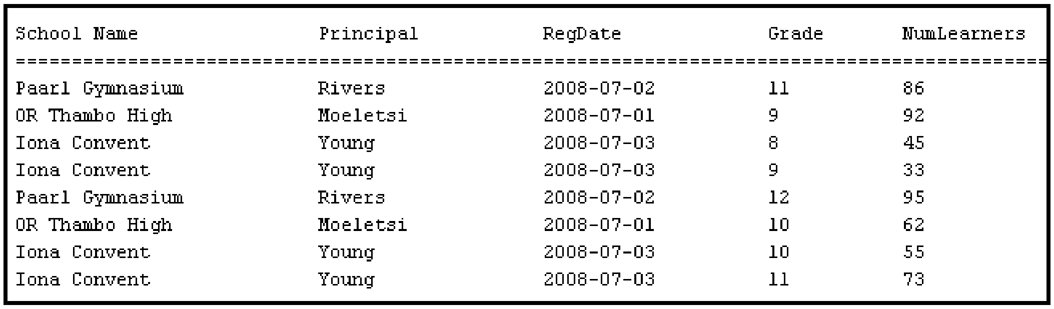
1.4 Complete the code in the **selectJuniors** method by formulating an SQL statement to determine the total weight of litter collected by the Grade 8 and 9 learners. Display the value as part of a user-friendly message. Example of the output:



```
Litter grade 8 and 9 learners collected in kg: 15193.2
```

(5)

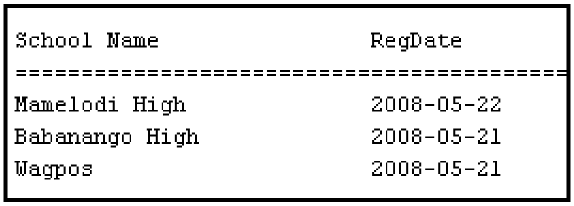
1.5 Complete the code in the **selectMonth** method by asking the user to enter the number of a month. Formulate an SQL statement to display the **Schoolname, Principal, Date Registered, Grade** and **Number of Learners** of all the schools that registered during the month entered by the user. Example of the output where the user enters 7 as the number of the month:



```
School Name               Principal          RegDate            Grade      NumLearners
==========================================================================================
Paarl Gymnasium           Rivers             2008-07-02         11         86
OR Thambo High            Moeletsi           2008-07-01         9          92
Iona Convent              Young              2008-07-03         8          45
Iona Convent              Young              2008-07-03         9          33
Paarl Gymnasium           Rivers             2008-07-02         12         95
OR Thambo High            Moeletsi           2008-07-01         10         62
Iona Convent              Young              2008-07-03         10         55
Iona Convent              Young              2008-07-03         11         73
```

(6)

1.6 Complete the code in the **selectDates** method by formulating an SQL statement to list the **SchoolName** and **RegDate** fields of all the schools that registered on 21/05/2008 or 22/05/2008. Example of the output:

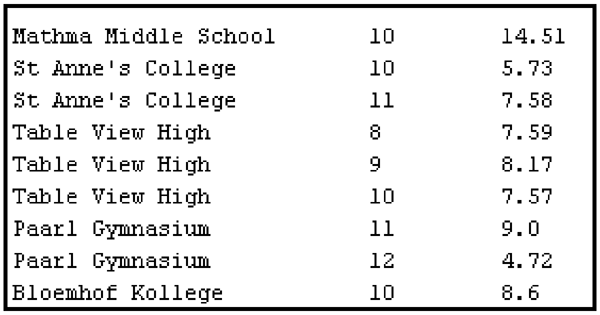


```
School Name                RegDate
==========================================
Mamelodi High              2008-05-22
Babanango High             2008-05-21
Wagpos                     2008-05-21
```

(5)

1.7 Complete the code in the **updateLearners** method by asking the user to enter the name of the school and the number of the learners to be added in the **SchoolsTb**. Formulate an SQL statement to add the number of learners to the school. (6)

1.8 Complete the code in the **calcLitterPerCapita** method by formulating an SQL statement to list the school name and a calculated field (named **PerCapita**) to calculate and display the average litter collected per learner in each grade. Also display the name of the school and the grade. NOTE: You will need to link the tables with an appropriate ***where*** clause to be able to do this. Example of output:



```
Mathma Middle School       10        14.51
St Anne's College          10        5.73
St Anne's College          11        7.58
Table View High            8         7.59
Table View High            9         8.17
Table View High            10        7.57
Paarl Gymnasium            11        9.0
Paarl Gymnasium            12        4.72
Bloemhof Kollege           10        8.6
```

:
: (5)

- Enter your examination number as a comment in the first line of the file named **Litter.java** containing the SQL statements.
- Save the **testLitter.java** and the **Litter.java** files.
- Rename the folder **Question 1 Java** to **Quest1_X**, where X should be replaced with your examination number.
- A printout for the code of the **Litter.java** file will be required.

[40]

QUESTION 2: JAVA – OBJECT-ORIENTED PROGRAMMING

This question is intended to test object-oriented programming skills. You are required to produce a solution that includes all classes specified in the instructions.

The amount of litter (in kilograms) collected by each school has been captured in a text file named **LitterWeight.txt** in the following format:

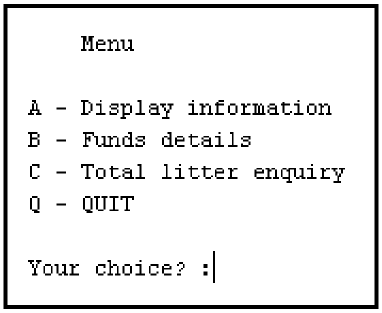
SchoolName#metal#glass#paper#other

An example of the data in the text file is:

```
Sutherland#56.5#12.2#95.0#32.4
Waterkloof#32.4#67.9#19.1#43.2
The Glen High#103.2#35.6#9.3#0.5
Riverside High#46.7#72.2#53.6#39.4
Soweto High#82.3#89.9#29.1#13.2
Babanango High#123.3#33.3#49.3#4.5
Four Ways#77.7#13.2#89.9#132.2
Paarl Gym#65.1#87.9#44.1#73.2
Driehoek#3.2#135.6#67.3#55.5
OR Tambo High#156.5#32.2#25.2#34.3
Eden College#12.1#127.9#67.1#23.2
Iona Convent#73.5#85.1#99.3#15.5
Mamelodi High#66.6#62.2#96.0#62.6
```

Do the following:

- Rename the folder **Question 2 Java** as **Quest2_X** (X represents your examination number).
- Open the **testSchoolLitter.java** file (class) in the folder **Quest2_X.**
- Change the options on the Menu so that it corresponds with the menu shown below.



```
    Menu

A - Display information
B - Funds details
C - Total litter enquiry
Q - QUIT

Your choice? :
```

- Add your examination number as a comment in the first line of the **testSchoolLitter** class.

2.1 Create an object class named **SchoolLitterXXXX** and save this unit as **SchoolLitterXXXX** in your **Quest2_X** folder. (XXXX should be replaced by the last four digits of your examination number.) Write the following code as part of this class:

2.1.1 This class must contain the following private fields:

fSchoolname
fMetalWeight
fGlassWeight
fPaperWeight
fOtherWeight

Ensure that you choose appropriate data types for these fields.
(4 ÷ 2) (2)

2.1.2 Write a parameterised constructor that will pass values for all the fields in the class. These parameters should be used to initialise the fields of the class. (4 ÷ 2) (2)

2.1.3 Write a method of type string named **toString** that returns information on a school in one string formatted as follows:

Name of school, Metal weight, Glass weight, Paper weight, Other weight

Example of return strings for the first school in the text file **LitterWeight.txt**:

The Glen High 123.6 kg 59.5 kg 203.4 kg 303.3kg 689.8 kg
(8 ÷ 2) (4)

2.1.4 Write a method named **totalWeight** which calculates the total weight of all the categories of waste collected by a school and returns a decimal value. (4 ÷ 2) (2)

2.1.5 Write a method named **fundsRaised** that accepts four decimal values as parameters indicating the amount paid per kilogram for each category of waste. The method works out how much money the school has made by multiplying the weights of the appropriate category with the value of that category (e.g. fPaperWeight * fPaperValue). (6 ÷ 2) (3)

2.1.6 Write accessor (get) methods for ALL of the fields in the class.
(6 ÷ 2) (3)

2.2 Do the following in the **testSchoolLitter** file in the given program:

2.2.1 Assign constant values to the four categories of weight in terms of rand per kilogram. Use the following values:

Metal waste:	R 1.50 per kg
Glass waste:	R 120 per kg
Paper waste:	R 1.00 per kg
Other waste:	R 0.50 per kg

(2 ÷ 2) (1)

2.2.2 Create an array named **arrSchools** that keeps objects of **SchoolLitterXXXX**. Write code to read information from the text file **LitterWeight.txt** according to the following steps:

(a) Test if the text file exists. If it does exist, open the text file and initialise a loop to read data. Display a suitable message if the file does not exist and terminate the program.

(b) Read a line of text from the text file.

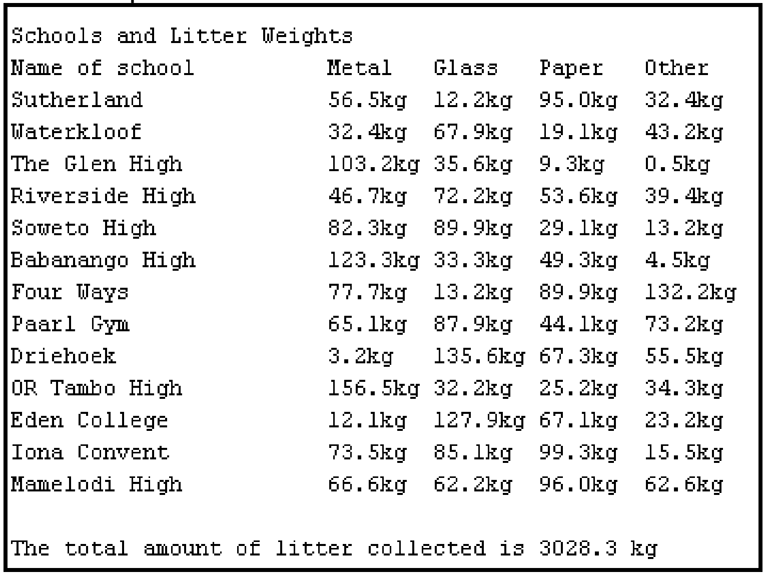
(c) Separate the text into the name of the school and the four weight figures indicating the kilograms of waste that have been collected by the school.

(d) Use this information to create a new **SchoolLitterXXXX** object and place the object into the array.

(e) Use a counter field to keep track of how many items there are in the array. (24 ÷ 2) (12)

2.2.3 Write code to complete the options on the menu provided in the program. The methods in the **SchoolLitterXXXX** class should be used where applicable. Invoke the relevant methods from the class.

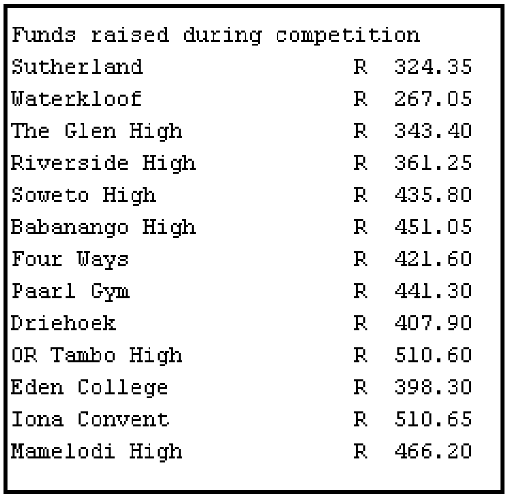
Display information: Display the names of the schools as well as the weights of the categories of litter they collected. Call the **toString** method to display the information. Display a suitable heading. Calculate and display the total amount of litter collected by the schools at the bottom of the list as indicated in the example of the output below:



```
Schools and Litter Weights
Name of school          Metal   Glass   Paper   Other
Sutherland              56.5kg  12.2kg  95.0kg  32.4kg
Waterkloof              32.4kg  67.9kg  19.1kg  43.2kg
The Glen High           103.2kg 35.6kg  9.3kg   0.5kg
Riverside High          46.7kg  72.2kg  53.6kg  39.4kg
Soweto High             82.3kg  89.9kg  29.1kg  13.2kg
Babanango High          123.3kg 33.3kg  49.3kg  4.5kg
Four Ways               77.7kg  13.2kg  89.9kg  132.2kg
Paarl Gym               65.1kg  87.9kg  44.1kg  73.2kg
Driehoek                3.2kg   135.6kg 67.3kg  55.5kg
OR Tambo High           156.5kg 32.2kg  25.2kg  34.3kg
Eden College            12.1kg  127.9kg 67.1kg  23.2kg
Iona Convent            73.5kg  85.1kg  99.3kg  15.5kg
Mamelodi High           66.6kg  62.2kg  96.0kg  62.6kg

The total amount of litter collected is 3028.3 kg
```

(10 ÷ 2) (5)

Funds raised: Display the detailed information on the funds raised by the schools during the waste collection competition. The program works out how much money the school has made by multiplying the weights of the appropriate category with the value of that category (e.g. fPaperWeight * fPaperValue). Display a suitable heading and subheadings. Example of the output: (on the next page).



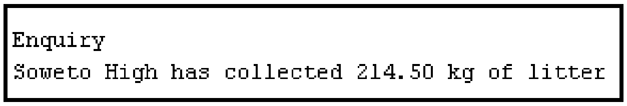
```
Funds raised during competition
Sutherland               R  324.35
Waterkloof               R  267.05
The Glen High            R  343.40
Riverside High           R  361.25
Soweto High              R  435.80
Babanango High           R  451.05
Four Ways                R  421.60
Paarl Gym                R  441.30
Driehoek                 R  407.90
OR Tambo High            R  510.60
Eden College             R  398.30
Iona Convent             R  510.65
Mamelodi High            R  466.20
```

(8 ÷ 2) (4)

Total litter enquiry: Allow the user to enter the name of a school. Code an effective way to search for the name in the array by stopping the loop as soon as the name has been found.

Display the name of the school and the total weight of litter collected by the school. Display a suitable message if the name was not found. The name of the school must be part of the message.

Example of output when the user input is Soweto High:



```
Enquiry
Soweto High has collected 214.50 kg of litter
```

(14 ÷ 2) (7)

- Enter your examination number as a comment in the first line of the **testSchoolLitter.java** file as well as the object class **SchoolLitterXXXX.java**.
- Save the files (File/Save All).
- Printouts of the code for the two files **testSchoolLitter.java** as well as **SchoolLitterXXXX.java** will be required.

[45]

QUESTION 3: JAVA PROGRAMMING

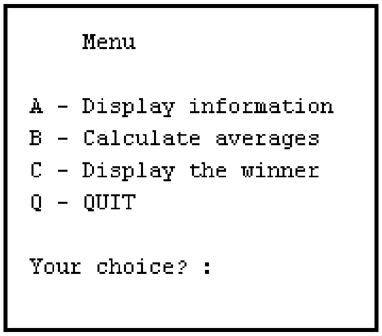
You are required to develop a program which summarises the results from the litter collection scheme run in 4 regions in a certain province for a number of schools. The number of schools participating in each region is not the same. The maximum number of schools that participated in a region was 6. The program should ask the user to enter the number of schools that participated in each region and then displays a simple text menu containing four options as indicated in FIGURE 3.1 below.

The program should generate random data to populate the array.

You have been supplied with an incomplete program in the folder named **Question 3 Java**.

Do the following:

- Rename the folder named **Question 3 Java** to **Quest3_X**. Replace X with your examination number.
- Open the Java program in this folder.
- Save the test class as **testRefuseCollectionXXXX** and the object class as **RefuseCollectionXXXX** inside the folder (XXXX should be replaced by the last four digits of your examination number).
- Change the **options** of the menu to match those of the menu below (FIGURE 3.1):

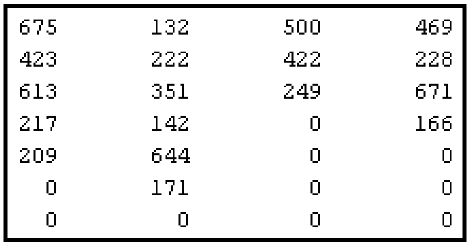


```
    Menu

A - Display information
B - Calculate averages
C - Display the winner
Q - QUIT

Your choice? :
```

FIGURE 3.1

3.1 Create methods to do the following:

3.1.1 Store the weight of litter collected by each school in a two-dimensional array. The array will hold data for 4 regions and a maximum of 6 schools in each region. An extra row must be provided to store the total weight of litter collected by the schools in a region.
For example – sample data:



```
675      132      500      469
423      222      422      228
613      351      249      671
217      142        0      166
209      644        0        0
  0      171        0        0
  0        0        0        0
```

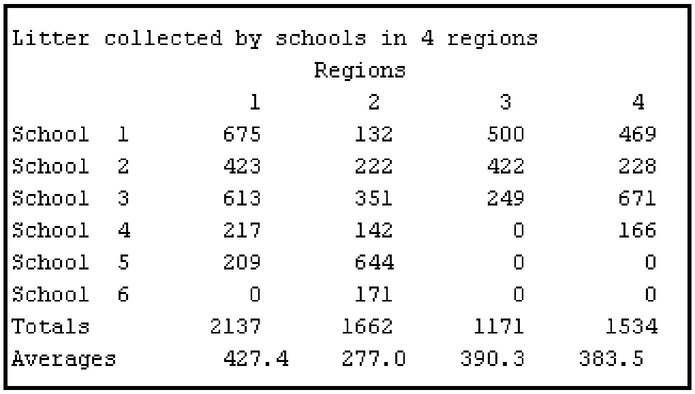
The array must be populated as follows:

- The program must request the user to input the number of schools in the first region which will be represented by the first column in the array. The weight of litter collected by each school is randomly generated and ranges from 100 to 800 kg and placed in the first column of the array.
- The same process will be repeated for the second, third and the fourth columns of the array. (18 ÷ 2) (9)

3.1.2 Calculate the total and the average weights of the litter collected by the schools in each region. Remember that each region may have a different number of schools. The average must be rounded off to one decimal place. (14 ÷ 2) (7)

3.1.3 Display the contents of the two-dimensional array with a suitable heading and suitable labels. Also display averages at the bottom of the array.

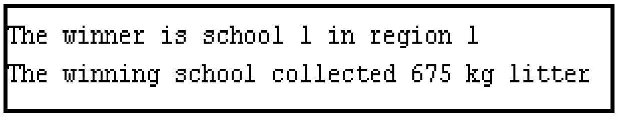
Example of output:



```
Litter collected by schools in 4 regions
                       Regions
                1        2        3        4
School  1     675      132      500      469
School  2     423      222      422      228
School  3     613      351      249      671
School  4     217      142        0      166
School  5     209      644        0        0
School  6       0      171        0        0
Totals       2137     1662     1171     1534
Averages        427.4    277.0    390.3    383.5
```

(20 ÷ 2) (10)

3.1.4 Determine and display the school that collected the most litter. Compile a message indicating the number of the school, the region where the school is situated and the weight of litter the school collected. Example of output:



```
The winner is school 1 in region 1
The winning school collected 675 kg litter
```

(14 ÷ 2) (7)

3.2 When the program is executed the two-dimensional array must be populated first before the menu in FIGURE 3.1 (see page 30) is executed.

For the option '**Display information**' on the menu you have to ensure that the code to be executed will display the contents of the array as well as the averages for the regions.

NOTE: The totals and averages will have values once the 'Calculate averages' option has been executed.

NOTE: The values will be different every time you run the program.

For the option '**Calculate averages**' you must ensure that the code to be executed will calculate the totals and the averages of the regions.

NOTE: If you want to view the output, execute the option 'Display Information' from the menu.

For the option '**Display the winner**' you must ensure that the code to be executed will determine and display the information on the winning school.

(4 ÷ 2) (2)

- Enter your examination number as a comment in the first line of the **testRefuseCollectionXXXX** and the **RefuseCollectionXXXX** classes.
- Save the two classes.
- Make printouts of the code of the **testRefuseCollectionXXXX** class and the **RefuseCollectionXXXX** class to hand in.

[35]

TOTAL SECTION B: 120

GRAND TOTAL: 120