



# basic education

Department:  
Basic Education  
**REPUBLIC OF SOUTH AFRICA**

**PINETOWN DISTRICT**

**GRADE 12**

**INFORMATION TECHNOLOGY P1**

**MID-YEAR EXAMINATIONS 2023**

**MARKS: 150**

**TIME: 3 hours**

**This question paper consists of 24 pages with 2 data pages.**

## INSTRUCTIONS AND INFORMATION

1. This paper is divided into FOUR sections. Candidates must answer ALL the questions in ALL FOUR 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. This question paper is set with programming terms that are specific to Delphi programming language. The Delphi programming language must be used to answer the questions.
4. Make sure that you answer the questions according to the specifications that are given in each question. Marks will be awarded according to the set requirements.
5. 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.
6. 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.
7. Routines, such as search, sort and selection, must be developed from first principles. You may NOT use the built-in features of the Delphi programming language for any of these routines.
8. All data structures must be defined by you, the programmer, unless the data structures are supplied.
9. You must save your work regularly on the disk/CD/DVD/flash disk you have been given, or on the disk space allocated to you for this examination session.
10. Make sure that your examination number appears as a comment in every program that you code, as well as on every event indicated.
11. If required, print the programming code of all the programs/classes that you completed. Your examination number must appear on all the printouts. You will be given half an hour printing time after the examination session.
12. At the end of this examination session, you must hand in a disk/CD/DVD/ flash disk 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 for this examination session. Ensure that all files can be read.

13. The files that you need to complete this question paper have been provided to you on the disk/CD/DVD/flash disk or on the disk space allocated to you. The files are provided in the form of password-protected executable files.

Do the following:

- Double click on the following password-protected executable file:  
**DataENGJun2023.exe**
- Click on the 'Extract' button.
- Enter the following password: **#MTHMY@1**

Once extracted, the following list of files will be available in the folder **DataENGJun2023**:

**Question1:**

Question1\_P.dpr  
Question1\_P.dproj  
Question1\_P.res  
Question1\_U.dfm  
Question1\_U.pas

**Question2:**

CollectionDB.mdb  
CollectionDB - Copy.mdb  
ConnectDB\_U.dcu  
Question2\_P.dpr  
Question2\_P.dproj  
Question2\_P.res  
Question2\_U.dfm  
Question2\_U.pas

**Question3:**

logo.jpeg  
Question3\_P.dpr  
Question3\_P.dproj  
Question3\_P.res  
Question3\_U.dfm  
Question3\_U.pas  
RecycleReceipt\_U.pas

**Question4:**

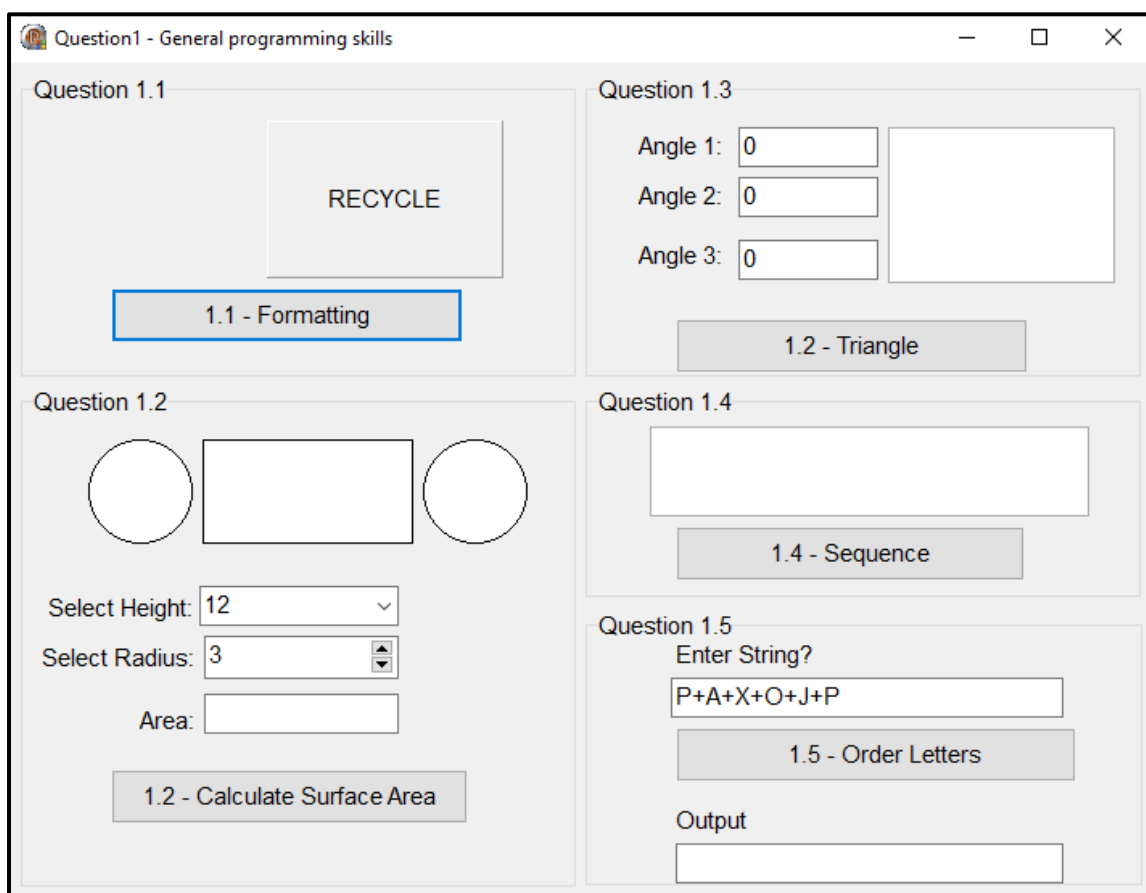
Quest4\_P.dpr  
Quest4\_P.dproj  
Quest4\_P.res  
Quest4\_U.dfm  
Quest4\_U.pas  
Updates.txt

**SECTION A****QUESTION 1: GENERAL PROGRAMMING SKILLS**

Do the following:

- Open the incomplete program in the **Question 1** folder.
- Enter your examination number as a comment in the first line of the **Question1\_U.pas** file.
- Compile and execute the program. The program has no functionality currently.

Example of graphical user interface (GUI):



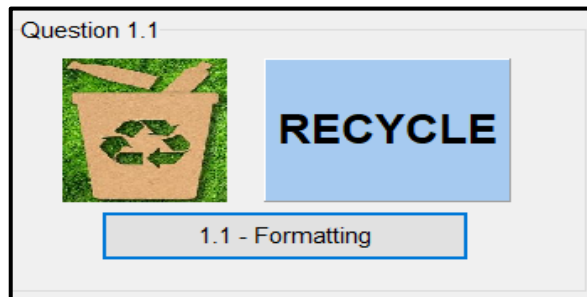
- Complete the code for each section of QUESTION 1, as described in QUESTION 1.1 to QUESTION 1.5 that follow.

### 1.1 Button [1.1 - Properties]

**Write code to** change the properties of the TImage (imgRecycle) and Panel (PnlSlogan) as follows:

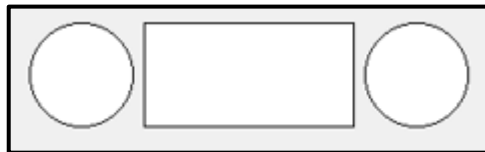
- The picture recycle.JPG must be put onto the imgRecycle component. Ensure that the entire picture fits in the tImage.
- In pnlSlogan, Set the font to Bold and the font-size to 20
- Change the color of the panel (pnlSlogan) to skyblue.

**Example of output:**



(5)

### 1.2 Button [1.2 – Calculate Surface Area]



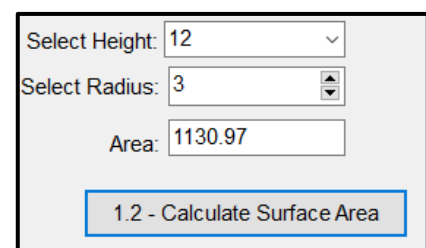
If you decompose the structure of a cylinder in order to find its surface area you will end up with two identical circles and a rectangle where the width of the rectangle is the circumference of the circle and the length of the rectangle is the height of the cylinder.

$$\text{Surface Area of cylinder} = 2\pi r^2 + 2\pi r h$$

Where:  $\pi = \text{Pi}$ ,  $r = \text{radius of circle}$  and  $h = \text{height of cylinder}$

**Write code to do the following:**

- Extract the radius and height of the cylinder from the given components.
- Calculate the surface area.
- Display the area on the richedit (edtQ1\_1\_2) component formatted to 2 decimal places.



(8)

### 1.3 Button [1.3 - Triangle]

Code to read the measurement of three angles of a triangle has been provided.

For a shape to be a triangle the sum of the angles must be  $180^\circ$

A triangle with 3 unequal angles is a SCALENE TRIANGLE

A triangle with 2 equal angles is an ISOSCELES TRIANGLE

A triangle with 3 equal angles is an EQUILATERAL TRIANGLE

**Write code to do the following:**

- Clear the richedit (redQ1\_3) component.
- Find the sum of the 3 angles.
- Use the information provided to check if the figure is a triangle
  - If the figure is a triangle, determine whether the triangle is SCALENE, ISOSCELES OR EQUILATERAL and output this information on the richedit (redQ1\_3) component.

If the figure is not a triangle, output the message 'Not a triangle' on the richedit (redQ1\_3) component.

**Example of output:**

|   |                      |
|---|----------------------|
| Angle 1: <input type="text" value="50"/><br>Angle 2: <input type="text" value="80"/><br>Angle 3: <input type="text" value="70"/><br><input type="button" value="1.3 - Triangle"/> | Not a Triangle       |
| Angle 1: <input type="text" value="50"/><br>Angle 2: <input type="text" value="70"/><br>Angle 3: <input type="text" value="60"/><br><input type="button" value="1.3 - Triangle"/> | Scalene triangle     |
| Angle 1: <input type="text" value="70"/><br>Angle 2: <input type="text" value="70"/><br>Angle 3: <input type="text" value="40"/><br><input type="button" value="1.3 - Triangle"/> | Isosceles Triangle   |
| Angle 1: <input type="text" value="60"/><br>Angle 2: <input type="text" value="60"/><br>Angle 3: <input type="text" value="60"/><br><input type="button" value="1.3 - Triangle"/> | Equilateral Triangle |

(10)

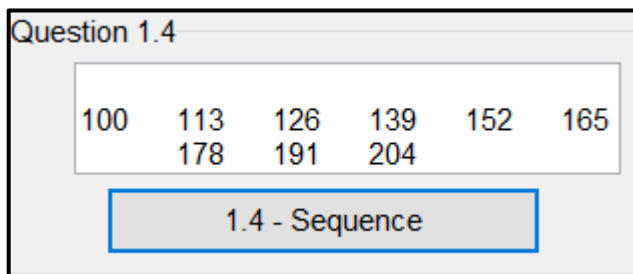
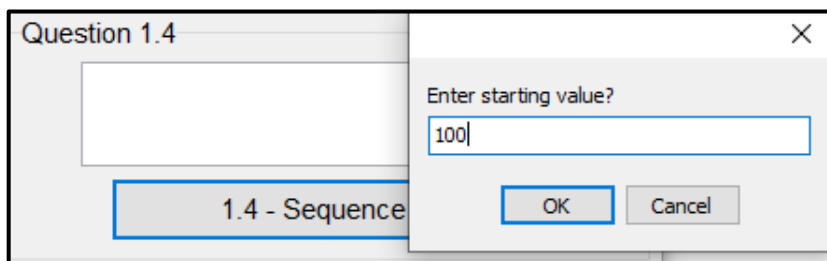
#### 1.4 Button [1.4 - Sequence]

A sequence must be formed and displayed by continuously adding 13 to a given number. This starting number is provided by the user. The sequence must stop as soon as a term of the sequence exceeds 200 for the first time.

#### Write code to do the following:

- Use an inputbox to get the starting value of the sequence.
- Use a conditional loop to continuously add 13 and output the term of the sequence. Terminate the loop when a term of the sequence exceeds 200 for the first time.
- Display the terms in the **redQ1\_4** output area.

Sample Output



(6)

#### 1.5 Button [1.5 – Order Letters]

A string of Capital letters separated by the '+' operator is provided as input in the edtQ1\_5\_1. The string starts with a capital letter and ends with a capital letter.

Write code to:

- Extract the word from the given edtQ1\_5\_1 component.
- Remove the '+' operators from the word.
- Process and display the word in **reverse alphabetical order** in the provided edtQ1\_5\_2 component.

Sample Output:

Question 1.5

Enter String?

Output

Question 1.5

Enter String?

Output

(11)

- Enter your examination number as a comment in the first line of the program file.
- Save your program.
- Print the code if required.

**TOTAL SECTION A: 40**



## SECTION B

### QUESTION 2: SQL AND DATABASE PROGRAMMING

The RECYCLE Company collects used cans and cardboard. They store their data in a database called **CollectionDB.mdb**, which contains information about the collection of cans and cardboard in kilograms from clients.

An application is required that will use the **CollectionDB.mdb** database to manage the data and payments to clients who participate in the recycling initiative.

The database contains two tables called **tblClients** and **tblCollection**.

The data pages attached at the end of the question paper provide information on the design of the **CollectionDB.mdb** database and its contents.

Do the following:

- Open the incomplete project file called **Question2\_P.dpr** in the **Question 2** folder.
- Enter your examination number as a comment in the first line of the **Question2\_U.pas** unit file.
- Compile and execute the program. The program has no functionality currently. The contents of the tables are displayed as shown below on the selection of tab sheet **Question 2.2 - Delphi code**.
- Follow the instructions below to complete the code for each section as described in QUESTION 2.1 and QUESTION 2.2.
- Use SQL statements to answer QUESTION 2.1 and Delphi code to answer QUESTION 2.2.

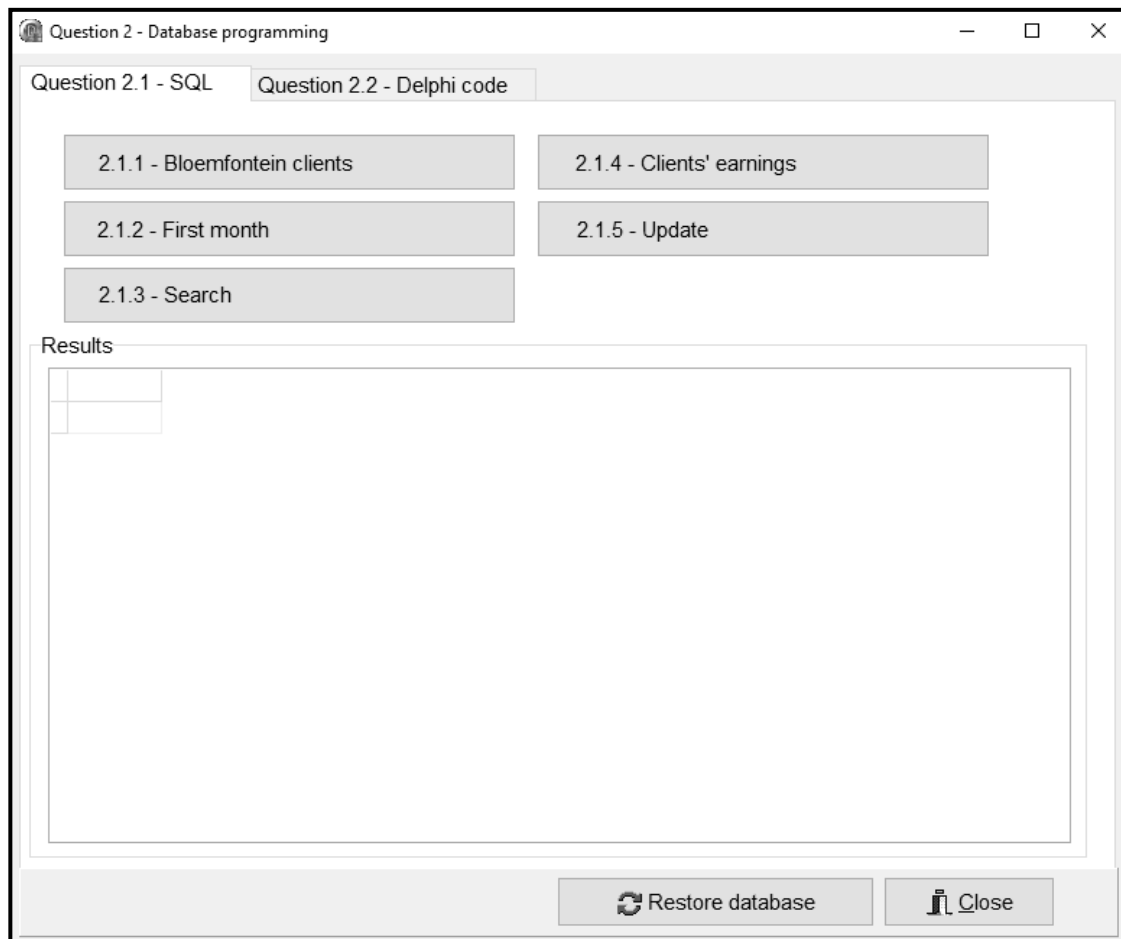
#### NOTE:

- The 'Restore database' button is provided to restore the data contained in the database to the original content.
- Code is provided to link the GUI components to the database. Do NOT change any of the code provided.
- Two variables are declared as public variables, as described below:

| Variable      | Data type | Description                              |
|---------------|-----------|--|
| tblClients    | TADOTable | Refers to the table <b>tblClients</b>    |
| tblCollection | TADOTable | Refers to the table <b>tblCollection</b> |

#### 2.1 Tab sheet [Question 2.1 - SQL]

Example of graphical user interface (GUI) for QUESTION 2.1:

**NOTE:**

- Use ONLY SQL code to answer QUESTION 2.1.1 to QUESTION 2.1.5.
- Code to execute the SQL statements and display the results of the queries is provided. The SQL statements that will be assigned to the variables **sSQL1**, **sSQL2**, **sSQL3**, **sSQL4** and **sSQL5** are incomplete.

Complete the SQL statements to perform the tasks described in QUESTION 2.1.1 to QUESTION 2.1.5 below.

**2.1.1 Button [2.1.1 – Durban clients]**

Display all details of clients who live in Durban from the **tblClients** table, sorted in Descending order of the **ClientSurname** field.

Example of output:

| ClientID | ClientName | ClientSurname | Address        | City   |
|----------|------------|---------------|----------------|--------|
| RHO09    | Rhoda      | Somers        | 14 Marilyn Way | Durban |
| PIE12    | Piet       | Mogorosi      | 5 Stormer Road | Durban |
| DAM07    | Damian     | Coetzer       | 12 Cape Avenue | Durban |

(4)

### 2.1.2 Button [2.1.2 - First month]

Display the **CollectionID**, **CollectionDate**, **NumberOfCans** and **KgsOfCardboard** of collections made in January, February and March that have NOT been paid.

Example of output of the first five records:

| CollectionID | CollectionDate | NumberOfCans | KgsOfCardboard |
|--------------|----------------|--------------|----------------|
| C002         | 2023/03/21     | 300          | 12             |
| C003         | 2023/03/23     | 250          | 18             |
| C005         | 2023/03/26     | 1200         | 78             |
| C006         | 2023/03/30     | 480          | 43             |
| C020         | 2023/03/11     | 650          | 32             |

(5)

### 2.1.3 Button [2.1.3 - Search]

Input the number of cans collected.

Display all details of records where the numberOfCans exceeds the input value and the CollectionDate is after the 20<sup>th</sup> of April 2023.

Example of output if the numberOfCans entered is 500:

| CollectionID | CollectionDate | NumberOfCans | KgsOfCardboard | ElectronicPayment | Paid  | ClientID |
|--------------|----------------|--------------|----------------|-------------------|-------|----------|
| C011         | 2023/04/29     | 1200         | 67             | True              | False | GER01    |
| C012         | 2023/05/08     | 800          | 87             | True              | False | ABI10    |
| C013         | 2023/05/19     | 625          | 76             | True              | True  | JOH03    |
| C015         | 2023/05/16     | 1000         | 90             | True              | True  | HEN11    |
| C016         | 2023/05/22     | 2400         | 128            | True              | True  | BUS06    |
| C024         | 2023/05/17     | 900          | 62             | True              | False | BUS06    |
| C028         | 2023/04/24     | 800          | 23             | False             | False | ABI10    |
| C053         | 2023/05/08     | 810          | 77             | False             | False | JOH03    |
| C059         | 2023/04/22     | 700          | 45             | True              | True  | RHO09    |
| C060         | 2023/05/21     | 900          | 76             | True              | True  | GER01    |
| C079         | 2023/04/30     | 2000         | 101            | True              | True  | BUS06    |
| C080         | 2023/05/05     | 3550         | 203            | True              | False | WIL12    |

(4)

### 2.1.4 Button [2.1.4 – Clients' earnings]

Clients receive R4,00 for every 5 kilogram of Cardboard collected.

A list of clients that have been paid is required.

Calculate and display the total amount that each client received for the Cardboard they have collected, formatted to currency. Display the **ClientName** field and the total amount received, using the field name **Income From Cardboard**.

Example of output of the first five records:

| ClientName | Income From Cardboard |
|------------|-----------------------|
| ▶ Busi     | R236.00               |
| Chris      | R604.80               |
| Damian     | R213.60               |
| Gert       | R144.80               |
| Henry      | R255.20               |

(8)

### 2.1.5 Button [2.1.5 - Update]

Many clients are opting for electronic payments.

Input the clientID.

Update the clientID to an electronic payment.

Code has been provided to display a message that indicates that a record has been changed in the database.

(4)

## 2.2 Tab sheet [Question 2.2 - Delphi code]

### NOTE:

- Use ONLY Delphi programming code to answer QUESTION 2.2.1 and QUESTION 2.2.2.
- NO marks will be awarded for SQL statements in QUESTION 2.2.

Example of graphical user interface (GUI) for QUESTION 2.2:

2.2.1 **Button [2.2.1 - Insert]**

Write code to add a new record to the **tblClients** table. The details of the client are as follows:

Client ID: **CHA01**  
 Client name: **Charles**  
 Client surname: **du Boit**  
 Address: **24 Van Wouw Street**  
 City: **Cape Town**

Example of the first four records of the **tblClients** table which shows that the record of the new client has been added successfully to the table:

| ClientID | ClientName | ClientSurname | Address             | City          |
|----------|------------|---------------|---------------------|---------------|
| CHA01    | Charles    | du Boit       | 24 Van Wouw Street  | Cape Town     |
| CHR08    | Chris      | Ferreira      | 188 Richmond Street | Potchefstroom |
| DAM07    | Damian     | Coetzer       | 12 Cape Avenue      | Durban        |
| GER01    | Gert       | Vermeulen     | 55 Dawn Street      | Bloemfontein  |

(4)

### 2.2.2 Button [2.2.2 - Percentage]

The company wants to calculate the total kilograms of cardboard collected by a specific client in a specific month as a percentage of the total kilograms of cardboard collected by the company in that specific month.

The user must do the following:

- Select a client from the DBGrid by clicking on the record.
- Select a month from the radio group **rgpQ2\_2\_2**.

Code has been provided to extract the month selected from the radio group **rgpQ2\_2\_2**.

Use the **redQ2\_2\_2** output area to display the information listed below.

Write code to do the following:

- Display the name and surname of the client selected.
- Determine and display the total kilograms of cardboard collected by the client for the month selected.
- Determine and display the total kilograms of cardboard collected by the RECYCLE company for the month selected.
- Calculate which percentage of the total kilograms of cardboard collected in the selected month, was collected by the client. Display the percentage formatted to two decimal places.

Example of output if the client record with ClientID **BUS06** and the February has been selected:

|                                 |      |
|---------------------------------|------|
| Busi Nkosi                      |      |
| Client collected in month 2:    | 66   |
| Company collected in month 2:   | 1357 |
| Percentage collected by client: | 4,86 |

Example of output if the client record with ClientID **CHR08** and the March has been selected:

|                                 |      |
|---------------------------------|------|
| Chris Ferreira                  |      |
| Client collected in month 3:    | 78   |
| Company collected in month 3:   | 1055 |
| Percentage collected by client: | 7,39 |

(11)

- Enter your examination number as a comment in the first line of the program file.
- Save your program.
- Print the code if required.

**TOTAL SECTION B: 40**

## SECTION C

### QUESTION 3: OBJECT-ORIENTATED PROGRAMMING

To assist with payments, the RECYCLE company requires a program that will create a receipt for clients.

Do the following:

- Open the incomplete program in the **Question 3** folder.
- Open the incomplete object class **RecycleReceipt\_U.pas**.
- Enter your name as a comment in the first line of both the **Question3\_U.pas** file and the **RecycleReceipt\_U.pas** file.
- Compile and execute the program. The program has limited functionality currently.

Example of graphical user interface (GUI):

Question 3 - Object-oriented programming

Question 3.2.1

Client ID:

Number of cans:

Kilograms of Cardboard::

3.2.1 – Instantiate object

Question 3.2.2

3.2.2 – Show Receipt

Question 3.2.3

Select number of Cans

3.2.3 – Update

Reset

- Complete the code as specified in QUESTION 3.1 and QUESTION 3.2 that follow.

**NOTE:** You are NOT allowed to add any additional attributes or user-defined methods, unless explicitly stated in the question.

- The provided incomplete object class TRceipt contains the declaration of four attributes:

The attributes for a **Receipt** object have been declared as follows:

| Attribute       | Description                    |
|-----------------|--------------------------------|
| fClientID       | A unique code for the client   |
| fNumberOfCans   | The number of cans collected   |
| fKgsOfCardboard | Cardboard in Kgs               |
| fAmount         | Total to be paid to the client |

Code has been provided for the following accessor methods:

- **getClientID** to return the fClientID attribute
- **getNumOfCans** to return the fNumberOf cans attribute
- **getKgsOfCardboard** to return the fKgOfCardboard attribute

Complete the code in the object class as described in QUESTION 3.1.1 to QUESTION 3.1.5 below.

- 3.1.1 Write code for a **constructor** method that will receive the ClientID the number of Cans and the kgsOfCardboard as parameters. Assign the parameter values to the respective attributes. Assign the default value of 0 to the fAmount attribute.

(5)

- 3.1.2 Write code for a method called **incNumCans** that receives a value parameter and increments the **fNumberOfCans** attribute by the value received.

(4)



3.1.3 Write code for a method called **calculateAmount** that will determine the total that will be paid to the client as follows:

- For every 5kgs of cardboard the client is paid R 4.
- The amount for cans is calculated as follows:

| Number of Cans      | Amount per Can |
|---------------------|----------------|
| 1500 cans and below | R 0,15         |
| 1501 - 2000         | R 0,20         |
| 2001 - 2500         | R 0,22         |
| Cans more than 2500 | R 0,30         |

(9)

3.1.4 Write code for a method called **setAmount** that receives an amount as a parameter which is used to set the fAmount attribute.

(2)

3.1.5 **Write a toString method to return a string with all the attributes of the object in the following format:**

Client ID: <ClientID>

Number of Cans: <NumberOfCans>

Kilograms of Cardboard: <KgsOfCardboard>

Total amount: <Amount>

(6)

**Amount must be formatted to currency.**

3.2 An incomplete program has been supplied in the **Question 3** folder. The program contains code for the object class to be accessible and declares an object variable called **objReceipt**.

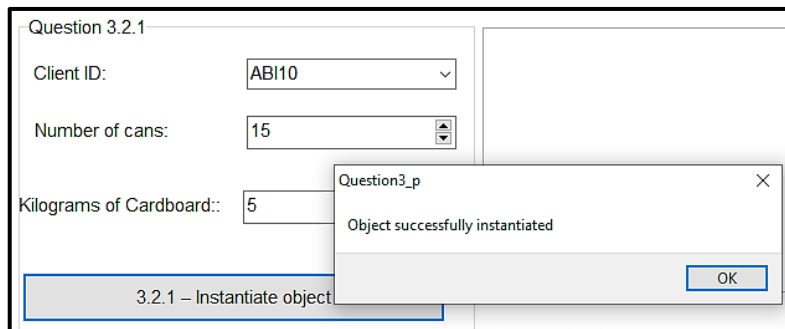
Write code to perform the tasks described in QUESTION 3.2.1 to QUESTION 3.2.4 below.

3.2.1 **Button [3.2.1 - Instantiate object]**

Write code to do the following:

- Extract the ClientID from the combo box **cmbQ3\_2\_1**, the number of cans from the spin edit **sedQ3\_2\_1\_Cans** and the Kilograms of Cardboard from the edit box **edtQ3\_2\_1\_Kgs**.

- Use the information to instantiate a new **Receipt** object.  
Display the message 'Object successfully instantiated'

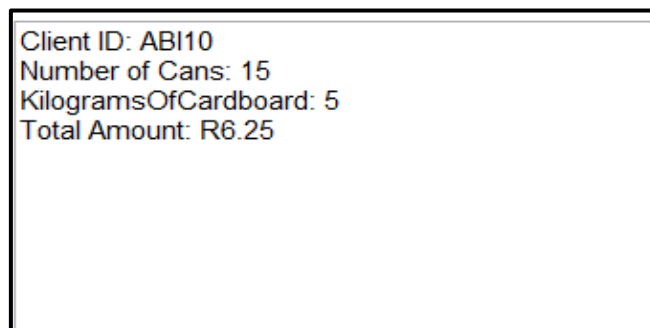


(6)

### 3.2.2 Button [3.2.2 – Show Receipt]

Display the receipt in the richedit **redQ3** component.

Example of output:



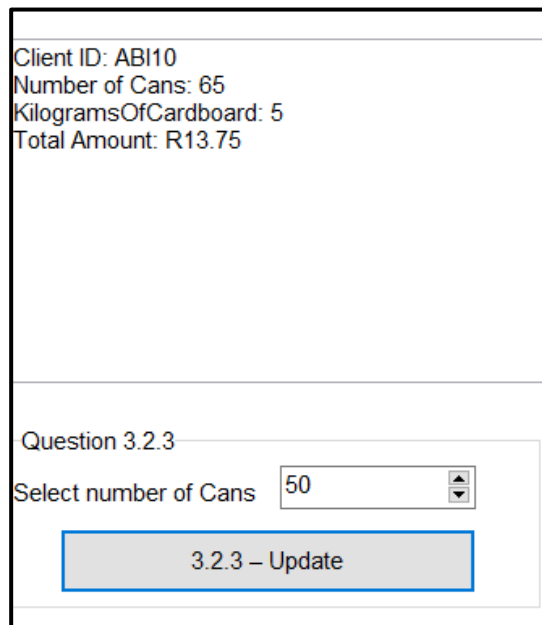
(2)

### 3.2.3 Button [3.2.3 - Update]

Write code to do the following:

- Extract the Extra Number of Cans from the **spinEdit (sedQ3\_2\_3)**.
- Call on the appropriate method to update the number of cans in the object class.
- Use the **toString** method to display the information of the updated **Receipt** object in the rich edit **redQ3**.

Sample output



(4)

- Enter your examination number as a comment in the first line of the program file.
- Save your program.
- Print the code if required.

**TOTAL SECTION C: 38**

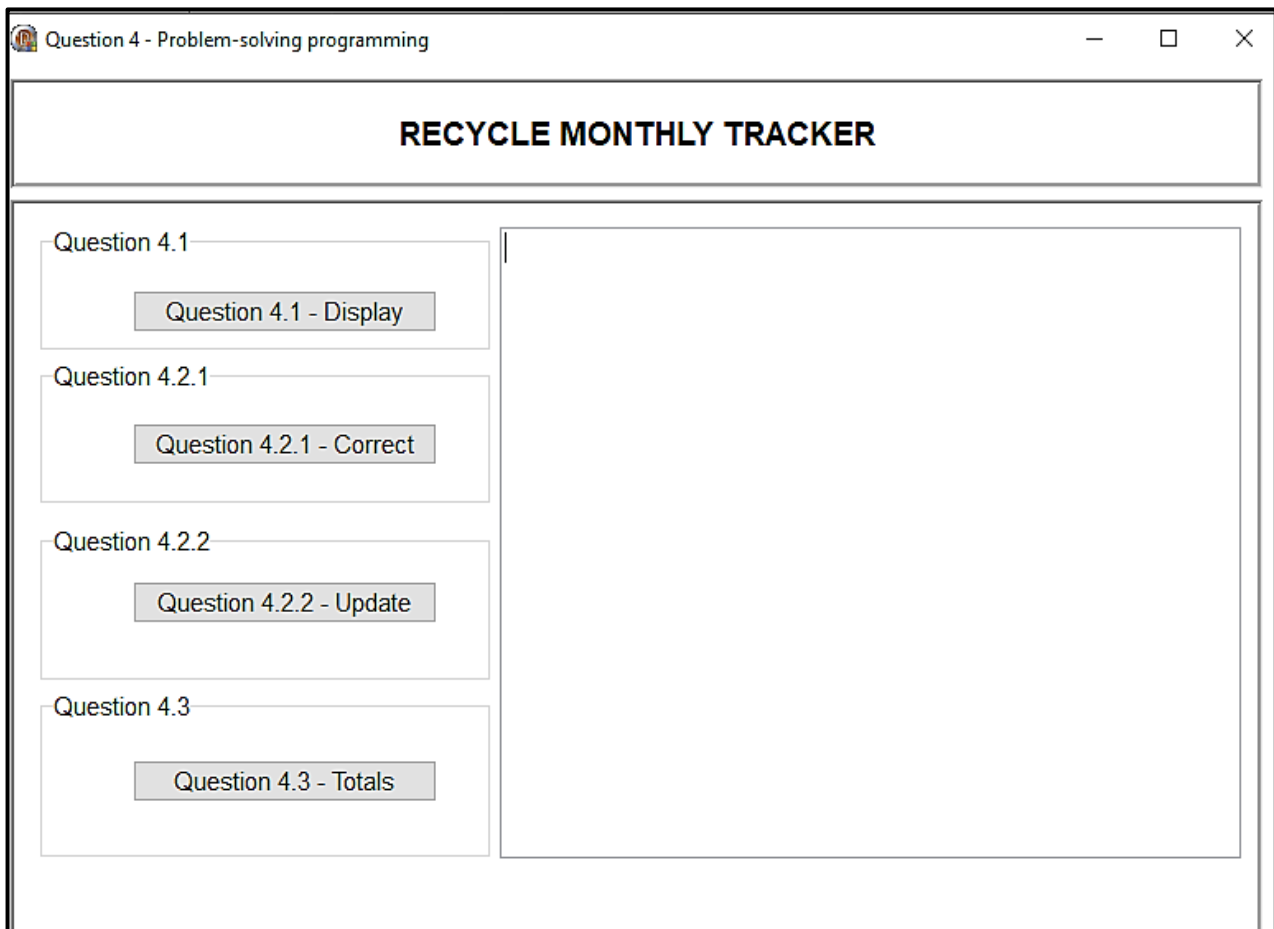
**SECTION D****QUESTION 4: PROBLEM-SOLVING PROGRAMMING**

Besides Metal and Cardboard, our RECYCLE company added, Paper and Plastic as new products that can be recycled. The company services 6 regions. A two-dimensional array is used to keep track of the quantity (in kilograms) received from each region.

Do the following:

- Open the incomplete program in the **Question 4** folder.
- Enter your examination number as a comment in the first line of the **Quest4\_U.pas** file.
- Compile and execute the program. The program has no functionality currently.

Example of graphical user interface (GUI):



The following arrays have been provided in the program:

- arrRegion: array [1 .. 6] of String = ('Durban', 'Verulam', 'Tongaat', 'Kwamashu', 'Inanda', 'Phoenix')
- arrCollection: array [1 .. 6, 1 .. 4] of Integer = ((23,12,82, 15), (5, 11, 65, 18), (19, 8, 72, 16), (28, 10, 68,12), (13, 9, 71, 17), (9, 11, 84, 11))

Complete the code for each section of QUESTION 4, as described in QUESTION 4.1 to QUESTION 4.3 below.

#### 4.1 Button [4.1 - Display]

The names of regions are provided in array **arrRegions**.

- The items in array arrCollection are integers that indicate the number of kilograms of each product that was received from each of the 6 regions.
- Code has been provided to display the heading.
- Write code to display the content of arrays **arrCollection** and **arrRegion** in neat rows and columns.

Example of output:

| Area     | Metal | Paper | Plastic | Cardboard |
|----------|-------|-------|---------|-----------|
| Durban   | 23    | 12    | 82      | 15        |
| Verulam  | 5     | 11    | 65      | 18        |
| Tongaat  | 19    | 8     | 72      | 16        |
| Kwamashu | 28    | 10    | 68      | 12        |
| Inanda   | 13    | 9     | 71      | 17        |
| Phoenix  | 9     | 11    | 84      | 11        |

(8)

4.2 It was noticed that there were errors in the data captured in the two-dimensional array.

#### 4.2.1 Button [4.2.1 - Correct]

The number of kilograms of cardboard collected for each of the 6 regions is entered under Plastics and vice-versa.

Write code to correct this problem.

Example of output:

| Area     | Metal | Paper | Plastic | Cardboard |
|----------|-------|-------|---------|-----------|
| Durban   | 23    | 12    | 15      | 82        |
| Verulam  | 5     | 11    | 18      | 65        |
| Tongaat  | 19    | 8     | 16      | 72        |
| Kwamashu | 28    | 10    | 12      | 68        |
| Inanda   | 13    | 9     | 17      | 71        |
| Phoenix  | 9     | 11    | 11      | 84        |

(5)

### Button [4.2.2 - Update]

A text-file called **Updates.txt** has been provided. Each line of the text-file consists of:  
**<first letter of the region>#<kgs of metal>#<kgs of paper>#<kgs of plastic>#<kgs of cardboard>**

The first 2 lines of the textfile:

```
D#2#4#9#2
T#13#4#16#2
```

The values in text-file must be used to update the appropriate values in the 2D array

Open the textfile, Updates.txt.

- Read each line and extract the first letter of the region, the no of kgs of metal,paper, plastic and cardboard.
- Add the new values to the existing values in the 2D array.
- Click on the button Display to show the updated figures.

### Sample output:

| Area     | Metal | Paper | Plastic | Cardboard |
|----------|-------|-------|---------|-----------|
| Durban   | 25    | 16    | 24      | 84        |
| Verulam  | 18    | 27    | 26      | 67        |
| Tongaat  | 32    | 12    | 32      | 74        |
| Kwamashu | 30    | 18    | 28      | 72        |
| Inanda   | 20    | 12    | 31      | 81        |
| Phoenix  | 24    | 18    | 23      | 92        |

(13)

**Button [4.3 – Totals ]**

Write code to determine the total number of kilograms for each of the different materials collected. Output the 2 D array and the totals below.

**Sample output:**

| Area     | Metal | Paper | Plastic | Cardboard |
|----------|-------|-------|---------|-----------|
| Durban   | 25    | 16    | 24      | 84        |
| Verulam  | 18    | 27    | 26      | 67        |
| Tongaat  | 32    | 12    | 32      | 74        |
| Kwamashu | 30    | 18    | 28      | 72        |
| Inanda   | 20    | 12    | 31      | 81        |
| Phoenix  | 24    | 18    | 23      | 92        |
|          | 149   | 103   | 164     | 470       |

(6)

- Enter your examination number as a comment in the first line of the program file.
- Save your program.
- Print the code if required.

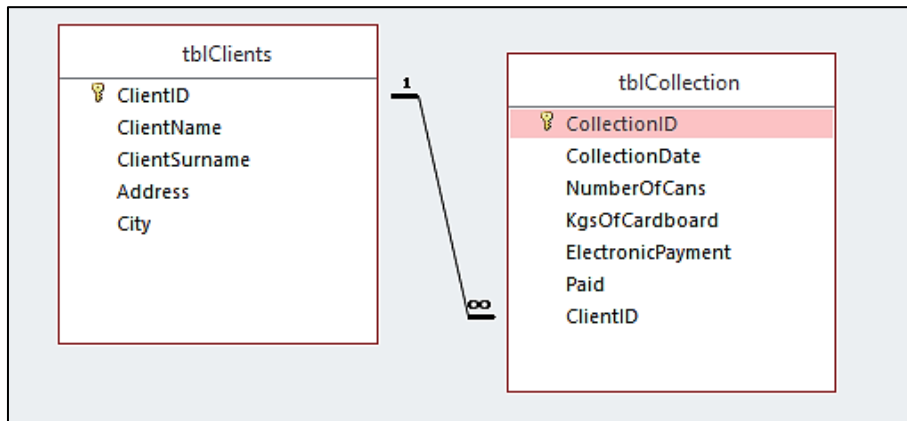
**TOTAL SECTION D: 32**  
**GRAND TOTAL: 150**

## INFORMATION TECHNOLOGY P1

### DATABASE INFORMATION QUESTION 2:

The database **CollectionDB** consists of table **tblClients** and **tblCollection**.

The following one-to-many relationship with referential integrity exists between the two tables in the database:



The design of the database tables is as follows:

#### Table: **tblClients**

This table contains details of the clients.

| Field name    | Data type | Description   |
|---------------|-----------|---|
| ClientID      | Text (5)  | Unique ID for the client                              |
| ClientName    | Text (15) | The name of the client                                |
| ClientSurname | Text (15) | The surname of the client                             |
| Address       | Text (20) | The address of the client used for the pickup of cans |
| City          | Text (15) | The city where the client resides                     |

Example of the records in the **tblClients** table:

| ClientID | ClientName | ClientSurname | Address             | City           |
|----------|------------|---------------|---------------------|----------------|
| ABI10    | Prashant   | Govender      | 72 Mountain Road    | Kimberley      |
| BUS06    | Busi       | Nkosi         | 65 Donald Road      | Welkom         |
| CHR08    | Chris      | Ferreira      | 188 Richmond Street | Potchefstroom  |
| DAM07    | Damian     | Coetzer       | 12 Cape Avenue      | Durban         |
| GER01    | Gert       | Vermeulen     | 55 Dawn Street      | Bloemfontein   |
| HEN11    | Henry      | Marques       | 1 Kingsley Drive    | Cape Town      |
| JAC05    | Jacob      | Human         | 8 Human Street      | Bloemfontein   |
| JOH03    | Johan      | Weston        | 43 Michellin Street | Vanderbijlpark |
| PHI04    | Phillip    | Brown         | 11 Park Road        | Sasolburg      |
| PIE12    | Piet       | Mogorosi      | 5 Stormer Road      | Bloemfontein   |
| RHO09    | Rhoda      | Somers        | 14 Marilyn Way      | Bloemfontein   |
| WIL12    | Willem     | de Wit        | 2 Arrow Street      | Sasolburg      |



Table: **tblCollection**

This table contains information of all the collections.

| Field name        | Data type | Description   |
|-------------------|-----------|---|
| CollectionID      | Text (5)  | Unique code for each collection   |
| CollectionDate    | Date/Time | Date of the collection  |
| NumberOfCans      | Number    | Integer value that indicates the number of cans collected                                       |
| KgsOfCardboard    | Number    | Integer value that indicates the weight of cardboard collected                                  |
| ElectronicPayment | Boolean   | A Boolean value that indicates if the client requires an electronic payment for the collection. |
| Paid              | Boolean   | A Boolean value that indicates that the client received payment for the cans collected          |
| ClientID          | Text (5)  | The ID of the client who collected the cans   |

Example of the first ten records in the **tblCollection** table:

| CollectionID | CollectionDate | NumberOfCans | KgsOfCardboard | ElectronicPayment                   | Paid                                | ClientID |
|--------------|----------------|--------------|----------------|-------------------------------------|-------------------------------------|----------|
| C001         | 2023/01/19     | 412          | 23             | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | WIL12    |
| C002         | 2023/03/21     | 300          | 12             | <input type="checkbox"/>            | <input type="checkbox"/>            | GER01    |
| C003         | 2023/03/23     | 250          | 18             | <input type="checkbox"/>            | <input type="checkbox"/>            | WIL12    |
| C004         | 2023/03/25     | 514          | 33             | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | WIL12    |
| C005         | 2023/03/26     | 1200         | 78             | <input type="checkbox"/>            | <input type="checkbox"/>            | CHR08    |
| C006         | 2023/03/30     | 480          | 43             | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | DAM07    |
| C007         | 2023/04/02     | 511          | 40             | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | GER01    |
| C008         | 2023/04/15     | 200          | 34             | <input type="checkbox"/>            | <input type="checkbox"/>            | BUS06    |
| C009         | 2023/04/17     | 419          | 22             | <input type="checkbox"/>            | <input type="checkbox"/>            | DAM07    |
| C010         | 2023/04/24     | 500          | 45             | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | CHR08    |

**NOTE:**

- Connection code has been provided.
- The database is password-protected, therefore you will not be able to access the database directly.