basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFORMATION TECHNOLOGY P1

NOVEMBER 2014 (2)

MARKS: 150

TIME: 3 hours

This question paper consists of 22 pages.

INSTRUCTIONS AND INFORMATION

1. This question paper is divided into THREE sections. Candidates must answer ALL THREE sections.

2. The duration of this examination is three hours. Because of the nature of this examination it is important to note that you will not be permitted to leave the examination room before the end of the examination session.

3. This paper is set in programming terms that are not specific to any particular programming language (Delphi/Java (using the Netbeans IDE)).

4. Make sure that you answer the questions according to the specifications that are given in each question. Marks will only 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 a programming language for any of these routines.

8. All data structures must be defined by you, the programmer. You may not use components provided within the interface to store and later retrieve data.

9. You must save your work regularly on the disk you have been given, or the disk space allocated to you for this examination.

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. 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 you need to complete this question paper have been given to you on a disk/CD/DVD/flash disk or on the disk space allocated to you in the form of a password-protected executable file:

- Delphi learners must use the file **DelphiDataENG.exe**.
- Java learners must use the file **JavaDataENG.exe**.

Do the following:

- Double click on the file.
- Click on the extract button.
- Enter the following password: **SocialNetworks@%**.

List of files provided in the folder DelphiDataENG/JavaDataENG (once extracted):

Delphi files

Question1:
EmailRecords.txt
Question1_P.dpr
Question1_P.res
Question1_U.dfm
Question1_U.pas
WelcomePict.bmp

Question2:
DataQ2.txt
Forum_U.pas
Question2_P.dpr
Question2_P.res
Question2_U.dfm
Question2_U.pas

Question3:
Blue crane.bmp
Question3_P.dpr
Question3_P.res
Question3_U.dfm
Question3_U.pas
Rhino.bmp

Java (Netbeans) files

Question1:
EmailRecords.txt
Question1.form
Question1.java
WelcomePict.png

Question2:
DataQ2.txt
Forum.java
Question2.form
Question2.java

Question3:
Blue crane.png
Rhino.png
PopulateArrays.java
Question3.form
Question3.java

SCENARIO

Social networking is an integral part of our lives and is a popular platform used by people all over the world for communication, shopping, research and many other useful purposes.

SECTION A

QUESTION 1: GENERAL PROGRAMMING SKILLS

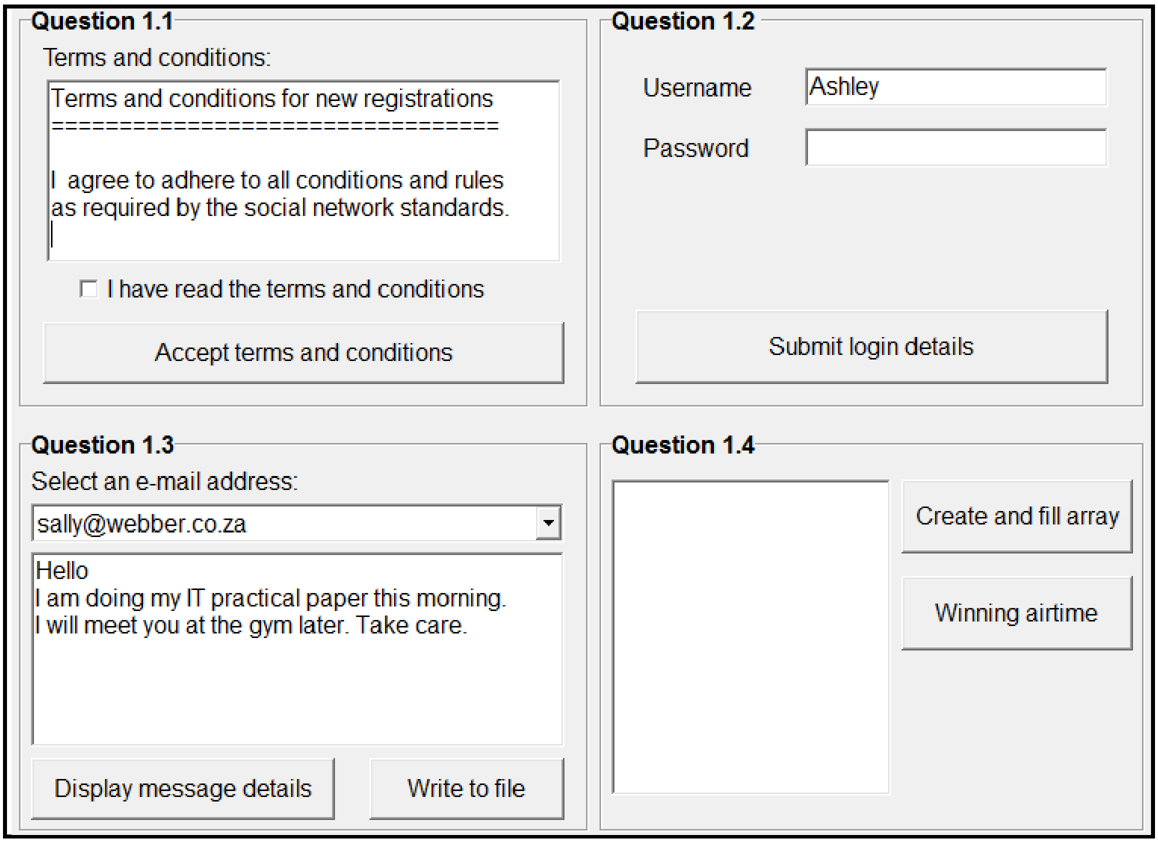
A social forum is available at your school for learners to interact and share ideas. To become a member of the forum learners need to accept the terms and conditions and then register by entering their username and password.

INSTRUCTIONS:

Delphi programmers	Java programmers
• The project **Question1** is provided in the **Delphi** folder.	• The project **Question1** is provided in the **Netbeans** folder.
• Open the incomplete project file **Question1_P.dpr** in the **Question1** folder.	• Open the incomplete class called **Question1.java** in the folders **Source Packages (src)** and **Question1Package**.
• Add your examination number as a comment in the first line of the main form unit **Question1_U.pas**.	• Add your examination number as a comment in the first line of the class **Question1.java**.

Do the following:

- Compile and execute the program. The interface displays four different sections labelled Question 1.1 to Question 1.4. Currently the program has no functionality. An example of the interface is shown on the next page.



Terms and conditions for new registrations
================================

I agree to adhere to all conditions and rules
as required by the social network standards.

□ I have read the terms and conditions

Username: Ashley

sally@webber.co.za

Hello
I am doing my IT practical paper this morning.
I will meet you at the gym later. Take care.

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

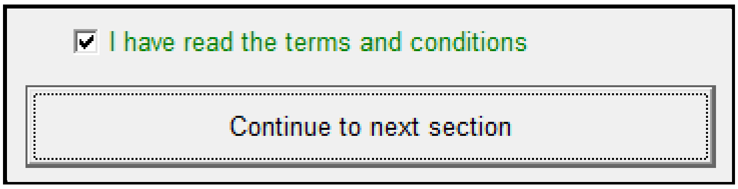
1.1 **Button – [Accept terms and conditions]**

The terms and conditions are displayed in the output area when the program is executed.

Write code to determine whether the check box has been selected.

- If the check box has been selected:
  - ○ Change the font colour of the text 'I have read the terms and conditions' to green.
  - ○ Change the caption/text of the button to 'Continue to next section'.
- If the check box is not selected:
  - ○ Change the font colour of the text 'I have read the terms and conditions' to red.

Example of output if the check box is selected:

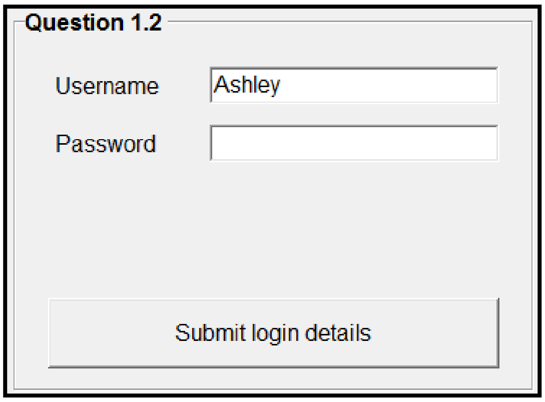


☑ I have read the terms and conditions

(4)

1.2 Button – [Submit login details]

The user must log in with a username and a password. A default username, 'Ashley', is given for testing purposes.



Ashley

Write code to test whether the password that was entered is valid or not.

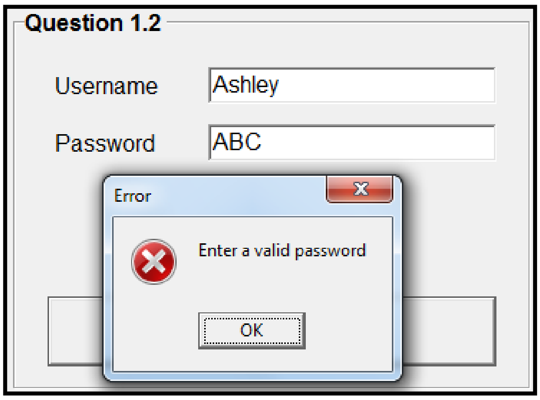
A **valid** password must contain the following:

- A minimum of eight characters
- At least one letter
- At least one number
- No spaces

If the password is **invalid**:

- Display a message 'Enter a valid password' in a dialog box.
- Clear the password text box.

Example of the output if the username is 'Ashley' and the password entered is 'ABC':



Ashley

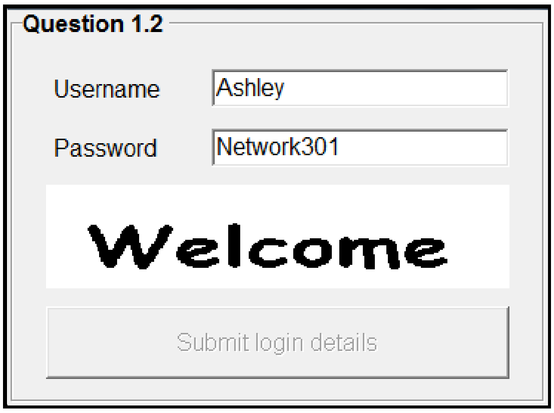
ABC

Enter a valid password

If the password is **valid**:

- Set the component that contains the 'Welcome' picture to visible. The component is currently invisible.
- Disable the 'Submit login details' button.

An example of the output if the username is 'Ashley' and the password is 'Network301':



Question 1.2

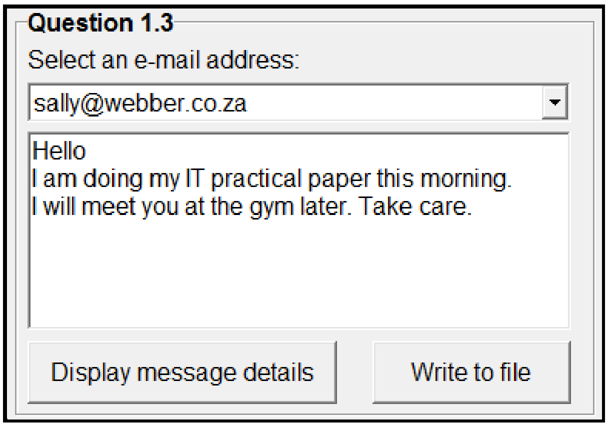
Username: Ashley

Password: Network301

(17)

1.3 Members of the forum can communicate by sending e-mail messages to other members on the forum. An e-mail message must be compiled by using the current date, the e-mail address of the recipient and the typed message.

The combo box provided contains the e-mail addresses of members of the forum. The user selects an e-mail address of a recipient from the combo box and enters a message in the output area provided.



Question 1.3

Select an e-mail address: sally@webber.co.za

Hello
I am doing my IT practical paper this morning.
I will meet you at the gym later. Take care.

1.3.1 **Button – [Display message details]**

Write code to do the following:

- Obtain the current date from the system.
- Create a string that contains the current system date, the e-mail address of the recipient and the message from the output area.

  Use the following format:

```
<Current date in the format YYYY-MM-DD>
<E-mail address of recipient>
<Open line>
<Message>
```

- Replace the original message in the output area with the newly created string.

Example of output if the current date is 2015-02-21, the e-mail address selected is benny@link.co.za and the message shown in the previous screenshot is used:

```
2015-02-21
benny@link.co.za

Hello
I am doing my IT practical paper this morning.
I will meet you at the gym later. Take care.
```

(8)

1.3.2 **Button – [Write to file]**

A text file called **EmailRecords.txt** has been provided which contains dates and e-mail addresses of the recipients of e-mails sent previously.

NOTE: The number of messages contained in the text file is unknown.

The format of each line in the text file is:

```
<date e-mail was sent>#<e-mail address of the recipient>
```

Example of the first six lines of the text file:

```
2014-04-18#harry@Kite.co.za
2014-04-23#Sally@mweb.co.za
2014-04-29#Mandy@cyber.co.za
2014-05-01#Author@space.co.za
2014-05-03#Kendra@myhouse.co.za
2014-05-15#Peggy@cyber.co.za
```

When a new e-mail is created an entry must be added to the **EmailRecords.txt** text file.

Write code to do the following when the '**Write to file**' button is clicked:

- Compile a string in the required format and write the string to the text file.
- Display a suitable message indicating that the information has been written to the file. (7)

1.4 You are required to organise a competition in which one can win airtime using social media.

1.4.1 **Button – [Create and fill array]**

Write code to do the following:

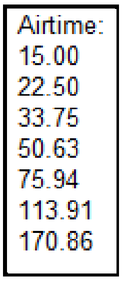
- Create a global array called **arrAirtime** that can save seven decimal values. The values to be placed in the **arrAirtime** array represent airtime amounts.

- Use loops to populate the array as follows:

  - The first amount in the array must be the value 15.
  - Each next value placed in the array must be 50% higher than the previous value in the array.

- Display the generated values in the given output area.

Example of output:



```
Airtime:
15.00
22.50
33.75
50.63
75.94
113.91
170.86
```

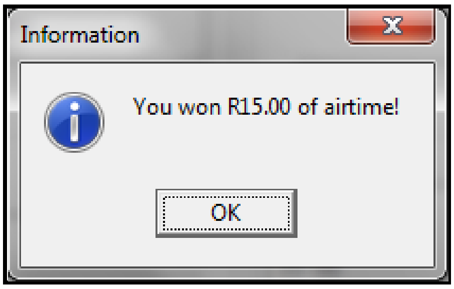
(8)

1.4.2 **Button – [Winning airtime]**

Write code to do the following:

- Randomly select an amount from the array.
- Display a message which contains the selected amount.

Example of output if the first position in the array was randomly generated:



Information

You won R15.00 of airtime!

NOTE: Due to the nature of the random function, the value displayed may differ each time the program is executed. (4)

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

TOTAL SECTION A: 48

SECTION B

QUESTION 2: OBJECT-ORIENTATED PROGRAMMING

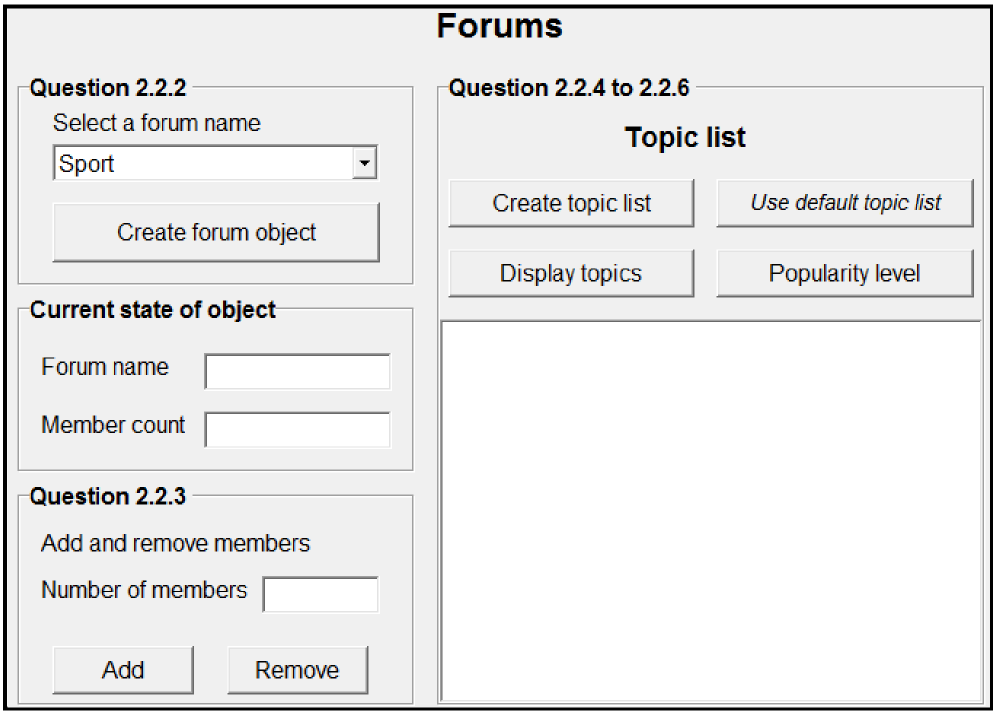
A company known as Communications Incorporated has created forums on common subjects such as sport, food, music, dance, animals and plants. You are provided with an incomplete object class called **Forum**. The information stored for each forum is the forum name, the number of members and the topics discussed in the forum.

INSTRUCTIONS:

Delphi programmers	Java programmers
• The project **Question2** is provided in the **Delphi** folder which contains: o A main form file called **Question2_U.pas** o An incomplete unit file called **Forum_U.pas** o A text file called **DataQ2.txt**	• The project **Question2** is provided in the **Netbeans** folder which contains: o A GUI class file called **Question2.java** o An incomplete object class called **Forum.java** o A text file called **DataQ2.txt**
• Open the project file called **Question2_P.dpr** in the **Question2** folder.	• Open the file called **Question2.java** in the folders **Source Packages (src)** and **Question2Package**.
• View ('View' (Ctrl+F12)) the unit file **Forum_U.pas** and add your examination number as a comment in the first line of files **Question2_U.pas** and **Forum_U.pas**.	• Add your examination number as a comment in the first line of the classes **Question2.java** and **Forum.java**.

Do the following:

- Compile and execute the program. Currently the program has no functionality. An example of the interface is shown on the next page.



- Complete the code as described in QUESTION 2.1 and QUESTION 2.2 to add functionality to the program.

2.1 The following class diagram shows the design of the **Forum** object class:

```
Forum
Attributes:
    -  forumName
    -  numMembers
    -  topicList
Methods:
    +  Constructor(forumName)
    +  getForumName()
    +  getNumMembers()
    +  getTopicList()
    +  addMembers(number)
    +  removeMembers(number)
    +  addTopic(topic)
    +  determinePopularityLevel()
```

Description of the attributes:

Names of attributes		
Delphi	**Java**	**Description**
fForumName	forumName	The name of the forum can be one of the following: Sport, Food, Music, Dance, Animals or Plants
fNumMembers	numMembers	The total number of members that use the forum
fTopicList	topicList	A list of topics discussed by forum members

NOTE: The attributes of the object have been declared as part of the given programming code.

2.1.1 Write code to create a **constructor** method to receive the forum name as a parameter. Initialise the forum name with the parameter value, set the number of members to 0 and initialise the topic list to a null value. (4)

2.1.2 Write code to create **accessor** methods for all three attributes. (3)

2.1.3 Write code to create a method called **addMembers** that will receive the number of new members as a parameter and will then add this value to the relevant attribute. (3)

2.1.4 Write code to create a method called **removeMembers** that will receive the number of members to be removed as a parameter and will then subtract this value from the relevant attribute. (3)

2.1.5 Write code to create a method called **addTopic** to receive a parameter containing a new topic. If the current topic list does not contain the new topic, then the new topic must be added to the list. (5)

2.1.6 Write code to create a method called **determinePopularityLevel** to return the word 'Low', 'Medium' or 'High'.

The popularity level depends on the number of members.

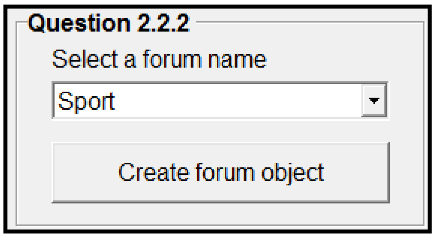
Number of members	Popularity level
From 0 up to 200	Low
Greater than 200 up to 500	Medium
Greater than 500	High

(8)

2.2 An incomplete interface class, **Question2_GUI**, is supplied. Open the class and follow the instructions given below.

2.2.1 Create a global object variable of the **Forum** class. The name of the variable must be **Forum** (Delphi)/**forum** (Java). (2)

2.2.2 Refer to the GUI below.



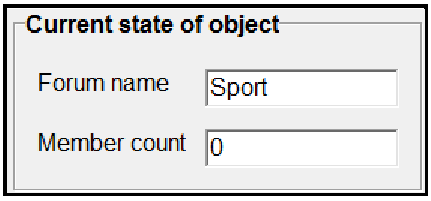
Select a forum name: Sport

Select a forum name from the combo box provided.

Button – [Create forum object]

- Use the forum name selected in the combo box to instantiate a forum object.
- Display the name and number of members (default 0) of the newly created forum in the **Forum name** and **Member count** text boxes provided.

Example of output for Sport forum selected and instantiated:

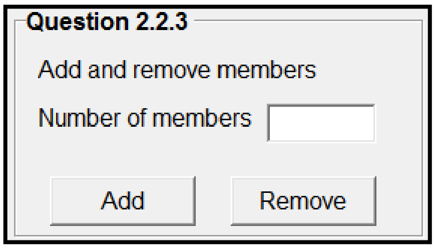


Forum name: Sport

Member count: 0

(6)

2.2.3 Refer to the GUI below.



The number entered in the text box represents the number of members to add to or remove from the forum.

(a) **Button – [Add]**

Write code to do the following:

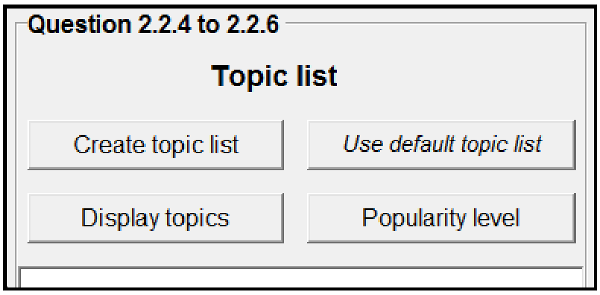
- Increase the number of members in the object by the value that was entered in the text box.
- Change the text box (**Member count**) in the GUI to show the updated number of members. (3)

(b) **Button – [Remove]**

Write code to do the following:

- Display the message 'Invalid input – type in new value' if the value entered is more than the number of members indicated by the member attribute of the forum object.
- If the number of members entered is valid, do the following:
  - ○ Decrease the member attribute in the forum object by the value that was entered in the text box.
  - ○ Display the updated number of members of the forum object in the text box (**Member count**). (5)

2.2.4 Refer to the GUI below.



Button – [Create topic list]

The text file called **DataQ2.txt** is supplied.

Each entry in the text file has the following format:

```
<Forum name>#<topic>
```

Example of the first seven lines of the text file is shown below.

```
Sport#Cricket
Music#Hip Hop
Dance#Ballroom
Food#Reality Shows
Animals#Herbivores
Plants#Indoor
Sport#Soccer
:
```

Write code to read the contents of the text file called **DataQ2.txt** as follows:

- Test whether the text file can be accessed. If the text file is not accessible, display the following message: 'File not found'.
- If the text file can be accessed, select the entries from the file that starts with the selected forum name. Add the topics specified in these entries to the topic list of the active forum object created in QUESTION 2.2.2.
- Display a suitable message in the output area, for example 'Topic list created successfully'.

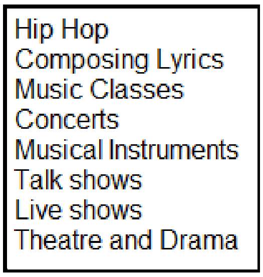
NOTE: If you were unsuccessful in writing code for this button,
OR
If the code does not produce the required results, you may do the following:

- Keep the code that you typed for the button, for marking purposes.
- Click on the button: [***Use default topic list***] to create a topic list that can be used for further testing. (13)

2.2.5 Button – [Display topics]

Write code to call the relevant method from the object class to display a list of topics for the object that was created in QUESTION 2.2.2.

Example of output for the Music forum is shown below.



```
Hip Hop
Composing Lyrics
Music Classes
Concerts
Musical Instruments
Talk shows
Live shows
Theatre and Drama
```

(2)

2.2.6 Button – [Popularity level]

Write code to call the relevant method from the object class to display the popularity level in the output area using the following format:

```
POPULARITY LEVEL: <popularity level>
```

Example of possible output if the forum has 501 members:

```
POPULARITY LEVEL: High
```

(2)

- Ensure that your examination number is entered as a comment in the first line of the class as well as the form.
- Save all the files.
- Print the code you created for both the classes, if printouts are required.

TOTAL SECTION B: 59

SECTION C

QUESTION 3: PROBLEM-SOLVING PROGRAMMING

The blue crane and the rhino are two species on the brink of extinction in South Africa. A fund-raising project is launched by using crowd funding. Crowd funding works on the principle that members of the public pledge an amount using social media.

The administrators of the project set a specific amount that needs to be collected. This amount is known as the **ultimate goal**. The ultimate goal for each of these projects has been set at R125 000. The ultimate goal value has been supplied as part of the given programming code.

Progress in the fund-raising project is monitored in order to determine when the project will be feasible.

Two parallel arrays with a maximum size of 20, called **arrConservationType** and **arrAmounts**, are given. The array **arrConservationType** stores the project code (**B** for the blue crane project or **R** for the rhino project). The array **arrAmounts** stores the pledge value that corresponds to the project type in array **arrConservationType**.

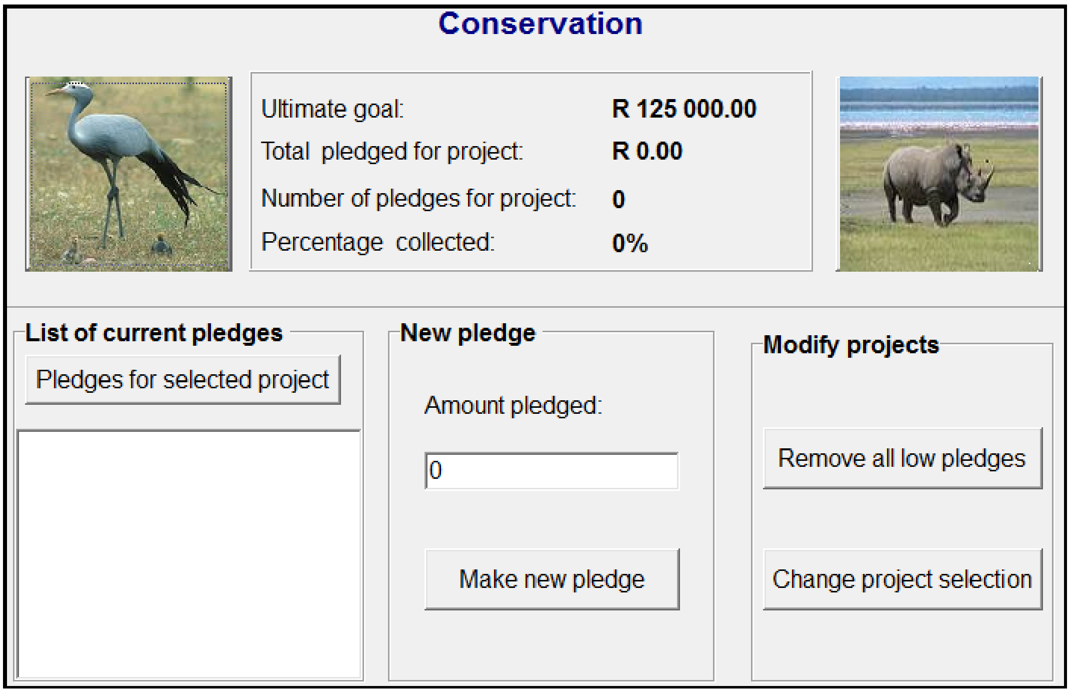
NOTE: These arrays have been populated with eleven initial pledges.

INSTRUCTIONS:

Delphi programmers	Java programmers
• The project **Question3** is provided in the **Delphi** folder.	• The project **Question3** is provided in the **Netbeans** folder.
• Open the incomplete project file **Question3_P.dpr** in the **Question3** folder.	• Open the incomplete class called **Question3.java** in the folders **Source Packages (src)** and **Question3Package**.
• Add your examination number as a comment in the first line of the main form unit file **Question3_U.pas**.	• Add your examination number as a comment in the first line of the class file **Question3.java**.

Do the following:

- Compile and execute the program. Currently the program has no functionality. An example of the interface is shown on the next page.



Ultimate goal: R 125 000.00
Total pledged for project: R 0.00
Number of pledges for project: 0
Percentage collected: 0%

Amount pledged: 0

NOTE: You are advised to use modular programming for an effective solution for this question.

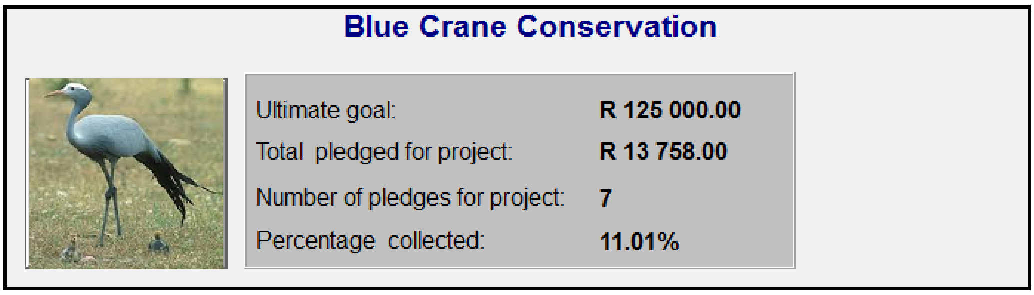
Complete the program to do the following:

3.1 **Buttons – [Blue Crane picture/Rhino picture]**

Two buttons are given, each with a picture of the conservation project that you can contribute to. When any one of the buttons is clicked, write code to do the following:

- Replace the heading 'Conservation' with either 'Rhino conservation' or 'Blue crane conservation', depending on the project that was selected.
- The button of the conservation project that was NOT selected must be hidden.
- Use the data in the supplied arrays to calculate and display the total amount pledged so far for the selected conservation project in the label provided.
- Display the percentage collected in comparison to the ultimate goal in the label provided.
- The background colour of the panel (pnlProject) depends on the percentage collected. The colour must be:
  - ○ Red if less than 40% of the ultimate goal is collected
  - ○ Yellow if 40% or higher of the ultimate goal is collected

Example of output if the blue crane is selected as the conservation project:



Blue Crane Conservation

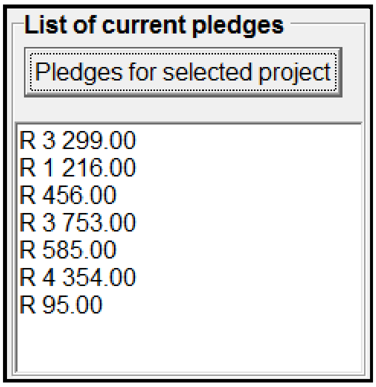
Ultimate goal: R 125 000.00
Total pledged for project: R 13 758.00
Number of pledges for project: 7
Percentage collected: 11.01%

(14)

3.2 **Button – [Pledges for selected project]**

Write code to display the current pledges made for the project selected in the given output area:

Example of output if the blue crane project is selected:



```
R 3 299.00
R 1 216.00
R 456.00
R 3 753.00
R 585.00
R 4 354.00
R 95.00
```

(4)

3.3 **Button – [Make new pledge]**

A new pledge must be received by allowing the user to enter the amount from the keyboard into the text box provided.

HINT: If you are unable to answer QUESTION 3.1, use the blue crane as the selected conservation project.

The following applies when a new pledge is made:

- A pledge will be successfully captured if:
  - ○ A project has been selected
  - ○ There is space in the arrays for the pledge
- If a pledge was successfully captured, do the following:
  - ○ Update the respective arrays with the selected project type and the value of the pledge.
  - ○ Update and display the amount pledged, the total number of pledges and the percentage collected in the respective components.

Use a dialogue box to display a user-friendly message to inform the user whether the pledge was accepted or not.

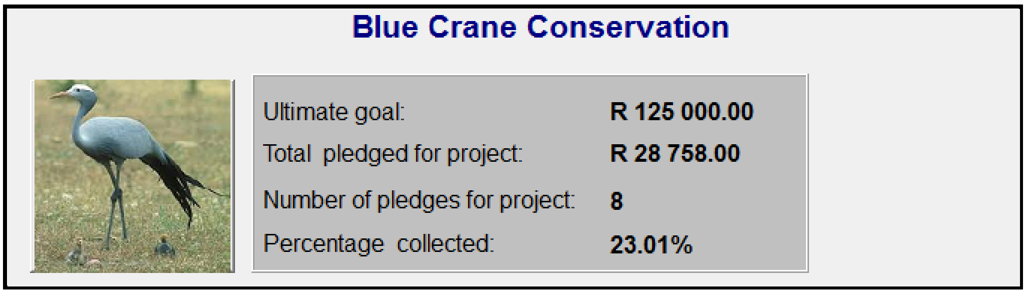
Example of output when a pledge was rejected:

```
Pledge has been unsuccessful.
```

Example of output when a new pledge was accepted:

```
Pledge has been registered.
```

The following is an example of the display after a successful pledge of R15 000 was made to the blue crane conservation project:



Blue Crane Conservation

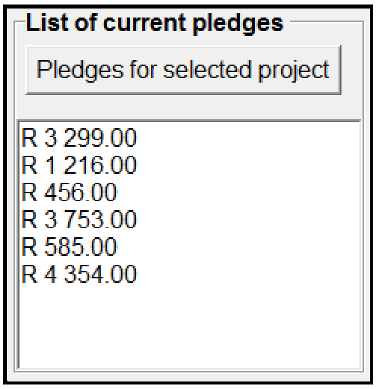
Ultimate goal: R 125 000.00
Total pledged for project: R 28 758.00
Number of pledges for project: 8
Percentage collected: 23.01%

(13)

3.4 **Button – [Remove all low pledges]**

High administration costs have made it uneconomical to collect pledges of less than R100. Write code to remove all pledges of less than R100 from both conservation projects.

Example of output of the list of pledges for the blue crane conservation project after removing all low pledges:



```
R 3 299.00
R 1 216.00
R 456.00
R 3 753.00
R 585.00
R 4 354.00
```

(12)

3.5 Button – [Change project selection]

The code for this button is supplied. The code resets the screen to its original state. Do NOT change any of the supplied code.

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

TOTAL SECTION C: 43
GRAND TOTAL: 150