basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFORMATION TECHNOLOGY P1

FEBRUARY/MARCH 2017

MARKS: 150

TIME: 3 hours

This question paper consists of 16 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 question paper is set with 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 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, 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. 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 given 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.

NOTE:

- Delphi candidates must use the file **DelphiDataENGMarch2017.exe**.
- Java candidates must use the file **JavaDataENGMarch2017.exe**.

Do the following:

- Double click on the password-protected executable file.
- Click on the extract button.
- Enter the following password: **Prof4U@#!**

Once extracted, the following list of files will be available in the folder **DelphiDataENGMarch2017/JavaDataENGMarch2017**:

DELPHI FILES

Question1:
Question1_P.dpr
Question1_P.dproj
Question1_P.res
Question1_U.dfm
Question1_U.pas
Type1.PNG
Type2.PNG

Question2:
CardHolder_U.pas
DataJanuary2017.txt
Question2_P.dpr
Question2_P.dproj
Question2_P.res
Question2_U.dfm
Question2_U.pas

Question3:
Question3_P.dpr
Question3_P.dproj
Question3_P.res
Question3_U.dfm
Question3_U.pas

JAVA (NETBEANS) FILES

Question1:
Question1.form
Question1.java
Type1.PNG
Type2.PNG

Question2:
CardHolder.java
DataJanuary2017.txt
Question2.form
Question2.java

Question3:
Question3.form
Question3.java

SCENARIO

Loyalty cards are often used to create profiles for customers. It is also a popular way of motivating customers to visit shops that regularly award loyalty points for purchases made.

SECTION A

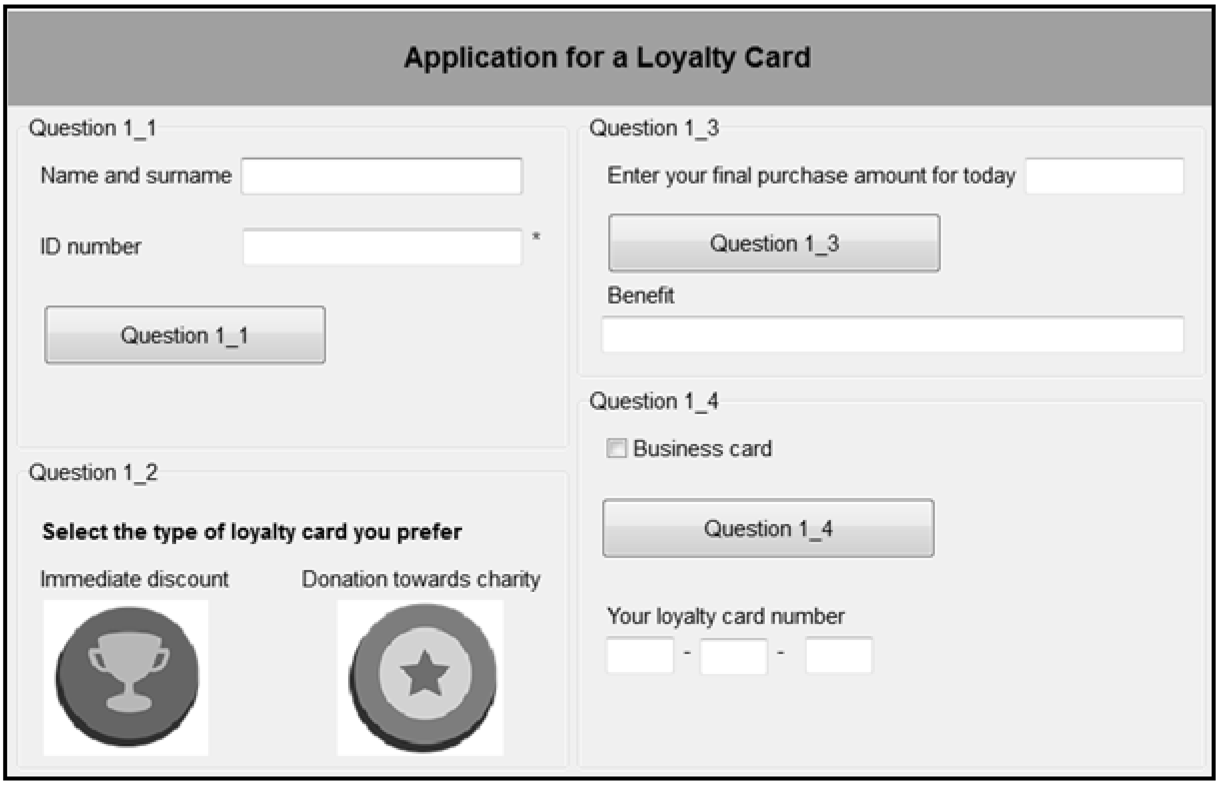
QUESTION 1: GENERAL PROGRAMMING SKILLS

An applicant must complete an application form to obtain a loyalty card.

Do the following:

- Compile and execute the program found in the **Question1** folder. The user interface displays four different sections named QUESTION 1_1 to QUESTION 1_4. Currently the program has no functionality.

  Example of graphical user interface (GUI):



- Complete the code for each section of QUESTION 1, as described in QUESTION 1.1 to QUESTION 1.4 on the next page.

1.1 Button [Question 1_1]

A label called **lblWelcomeMessage** has been created and set to be not visible. Write code to validate the information entered by the user as follows:

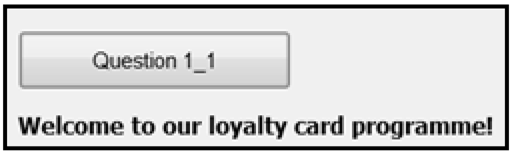
- Name and surname

The input must contain at least one space to be valid. If the input is not valid, display a message to indicate the specific error using a dialog box.

- ID number

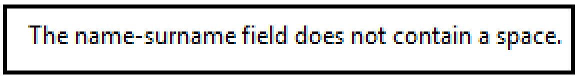
The asterisk character (*) next to the text box indicates that the field may not be left blank. An ID number must consist of 13 digits to be valid. If the ID number field does not contain any information or the ID number that was entered does not consist of 13 digits, display a message to indicate the specific error that occurred using a dialog box.

If the contents in both fields are valid, the label displaying the welcoming message must be set to visible.



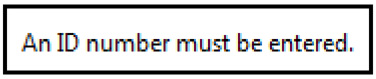
Welcome to our loyalty card programme!

Example of message if no space was entered in the name-surname field:



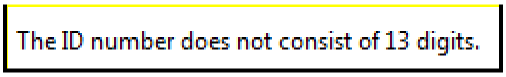
The name-surname field does not contain a space.

Example of message if the ID number field was left blank:



An ID number must be entered.

Example of message if the ID number that was entered does not consist of 13 digits:



The ID number does not consist of 13 digits.

(14)

1.2 Buttons with images [Question 1_2]

A customer must select the preferred type of loyalty card. The images on the buttons in this section of the GUI show two types of loyalty cards. When a specific image is selected (button clicked), the program must do the following:

- The heading in the label above the selected image must be changed to bold.

- A code for the type of loyalty card that was selected must be assigned to the global variable **typeCard** that is provided. The code DI represents immediate discount and the code CH represents a donation towards charity.

NOTE: Only the label for the currently selected button must appear in bold. If the user selects the other type of loyalty card, the label of the previously selected loyalty card must NOT appear in bold. (6)

1.3 **Button [Question 1_3]**

The final purchase amount for today must be entered in the text box.

The program must grant the following benefit towards the holder of the type of loyalty card selected in QUESTION 1_2:

- Immediate discount (DI):

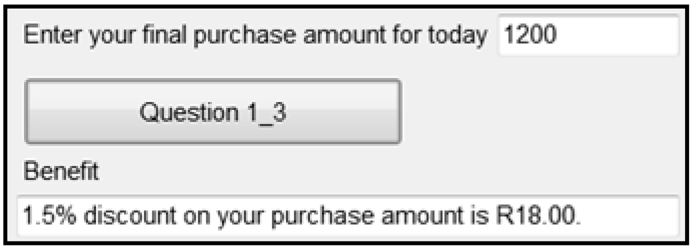
  1,5% discount must be granted on the purchase amount.

- Donation towards charity (CH):

  An amount of R3 will be donated to a charity organisation for every R100 or part of R100 spent by the loyalty card holder.

Display a message describing the benefit granted depending on the type of loyalty card.

Example of output if the code of the loyalty card is DI and the purchase amount R1 200:



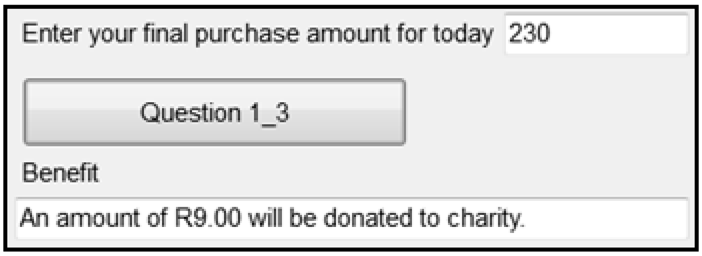
Enter your final purchase amount for today 1200

Benefit

1.5% discount on your purchase amount is R18.00.

NOTE: The discount amount must be displayed as a currency with two decimal digits.

Example of output if the code of the loyalty card is CH and the purchase amount R230:



Enter your final purchase amount for today 230

Benefit

An amount of R9.00 will be donated to charity.

(15)

1.4 **Button [Question 1_4]**

The customer's ID number entered in QUESTION 1_1 and the type of loyalty card selected in QUESTION 1_2 are required to compile the customer's loyalty card number. A loyalty card number consists of three parts. Each part of the number must be compiled as follows and displayed in the components provided:

Part 1:

The first part of the loyalty card number consists of either the letter B if the business card check box was selected or the letter P if the business card check box was not selected, followed by the first three digits of the customer's ID number.

Part 2:

A string called **charString**, which contains the following 10 characters, has been supplied as part of the code provided:

```
! @ # $ % ] & * ^ ~
```

Each character contained in the string represents a digit (0–9). This means that the first character (!) represents the digit 0, the second character (@) the digit 1, and so on.

This part of the loyalty card number must consist of the last digit from the customer's ID number, followed by the character from **charString** that represents that digit.

Example of this part of the loyalty card number if the customer's ID number is 8802128756084:

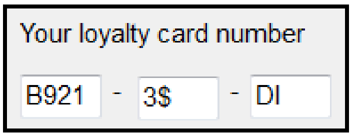
4%

Part 3:

This part of the loyalty card number consists of the code representing the type of loyalty card (DI or CH) established in QUESTION 1_2.

Example of the loyalty card number of a customer with the following profile:

- Application for a business
- ID number: 9210140012083
- Card type: DI



Your loyalty card number

B921 - 3$ - DI

(13)

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

TOTAL SECTION A: 48

SECTION B

QUESTION 2: OBJECT-ORIENTATED PROGRAMMING

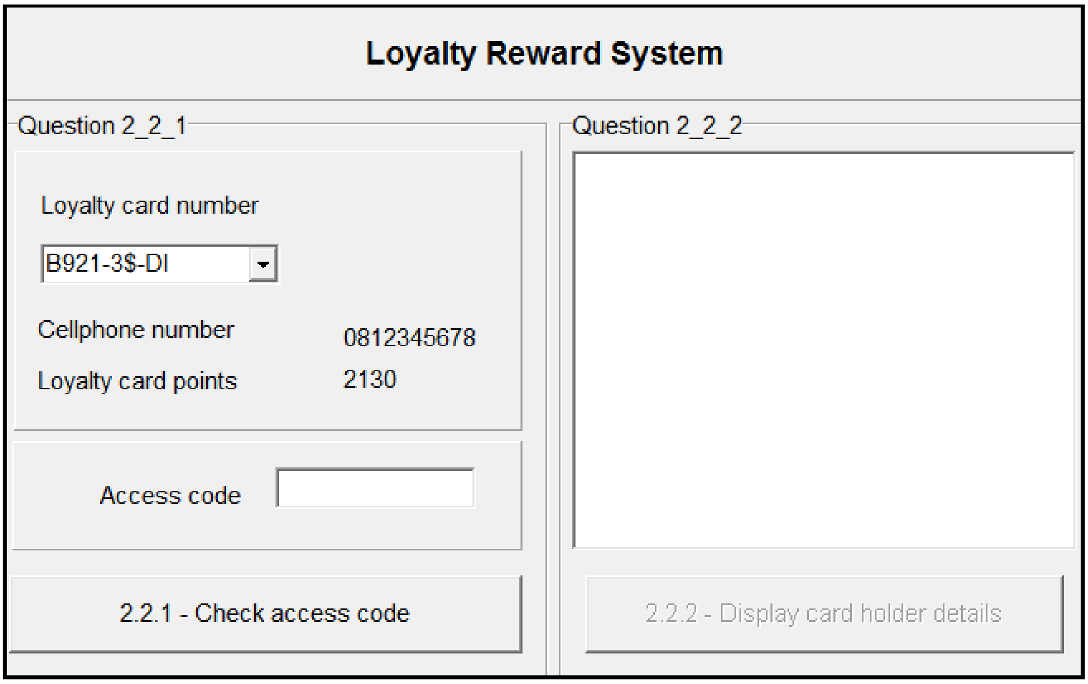
The local supermarket has a loyalty card system where customers can receive rewards depending on the number of visits, the number of loyalty points gathered and the percentage of health food bought.

Do the following:

Delphi programmers	Java programmers
• Open the incomplete program in the **Question2** folder. • Open the incomplete object class **CardHolder_U.pas**. • Add your examination number as a comment in the first line of both files **Question2_U.pas** and **CardHolder_U.pas**.	• Open the incomplete program in the **Question2** folder. • Open the incomplete object class **CardHolder.java**. • Add your examination number as a comment in the first line of both classes **Question2.java** and **CardHolder.java**.

- Compile and execute the program. Currently the program has no functionality.

  Example of graphical user interface (GUI):



Loyalty Reward System

Loyalty card number: B921-3$-DI

Cellphone number: 0812345678

Loyalty card points: 2130

Question 2_2_1
Question 2_2_2
Access code
2.2.1 - Check access code
2.2.2 - Display card holder details

- Complete the code for this program, as specified in QUESTION 2.1 and QUESTION 2.2 below.

2.1 The given incomplete object class (**TCardHolder/CardHolder**) contains the declaration of five attributes, the **toString()** method and an incomplete **isCorrect()** method.

The given attributes of the **CardHolder** object:

NAMES OF ATTRIBUTES		DESCRIPTION
Delphi	**Java**	
fCardNumber	cardNumber	A unique number for the card holder
fCellNumber	cellNumber	The cellphone number of the card holder
fNumVisits	numVisits	The number of times the card holder visited the shop during the month
fLoyaltyPoints	loyaltyPoints	The total number of loyalty points received on this card up to now
fHealthLevel	healthLevel	A character indicating the health level of the card holder, based on the amount of health food bought during the month: S – Silver (Lowest level) G – Gold P – Platinum (Highest level)

Complete the code in the given **CardHolder** class (**TCardHolder/CardHolder**) as described in QUESTION 2.1.1 to QUESTION 2.1.6 below.

2.1.1 Write code for a **constructor** method to receive the card number, cellphone number and loyalty card points already received, as parameter values. Assign these values to the relevant attributes.

The lowest code for the health food level (S) must be assigned to the attribute for the health level. The attribute for the number of visits must be set to 0. (5)

2.1.2 Write a **mutator** method to be able to set the attribute for the number of visits to a value received as a parameter. (2)

2.1.3 Write code for a method called **increaseLoyaltyPoints** that receives the total amount spent for this month as a parameter and increases the current number of loyalty points based on the amount received.

Use the following criteria:

For every four rand (R4) spent, one loyalty point is awarded. (5)

2.1.4 Write code for a method called **updateHealthLevel** that will receive the total amount spent for this month and the amount spent on health food for this month as parameters. Calculate the percentage that was spent on health food and set the attribute for the health level according to the criteria provided in the table below.

Level code	Percentage indication
S	Less than 10%
G	10% or more, but less than 40%
P	40% or more

(7)

2.1.5 An access code is required to be able to update a customer's loyalty points. The customer's cellphone number has been used to compile the access code.

Write code to complete the **isCorrect** method provided. The method must test whether the access code received as a parameter is correct by applying the following instructions on the attribute for the cellphone number:

- Remove all zeros (0) from the cellphone number.

- Determine the sum of the remaining digits as follows:

  - If the number of remaining digits is even, every two consecutive digits starting on the left-hand side must be regarded as a two-digit value. Add all these two-digit values.

    Example:

    If the original cellphone number was 082 340 2113, the remaining digits without any zeros would be 82342113. The sum of the two-digit values is:

    82 + 34 + 21 + 13 = 150

  - If the number of remaining digits is odd, the first digit must be regarded as a one-digit value, and thereafter every two consecutive digits must be regarded as a two-digit value. Add all these values.

    Example:

    If the original cellphone number was 082 034 2101, the remaining digits after removing the zeros are 8234211. The sum of the values is:

    8 + 23 + 42 + 11 = 84 (13)

2.1.6 Write code for a method called **identifyStarShopper** that will return a string to indicate whether the card holder is a STAR shopper or not.

A card holder is a STAR shopper if the following applies:

- The card holder earned more than 2 000 loyalty points and the shop was visited more than 10 times during the month.
  OR
- The health level of the card holder is platinum.

The string to be returned must either be 'STAR shopper' or an empty string. (6)

2.2 Information on the visits and purchases of card holders is saved for one month only in a text file. The text file called **DataJanuary2017.txt** contains the transactions made by four of the card holders during January.

The contents of text file is as follows:

Every three lines represent a transaction for a specific loyalty card holder:

Line 1: The card number of the card holder, for example B921-3$-DI
Line 2: The total amount of the purchases for this visit, for example 1459.30
Line 3: The amount spent on health food during this visit, for example 72.30

Example of the first 12 lines of the text file **DataJanuary2017.txt**:

```
B921-3$-DI
1459.30
72.30
B870-4%-CH
899.80
22.50
P730-7*-DI
299.30
62.85
B870-4%-CH
869.30
182.55
```

Do the following to complete the code for each button in the main form unit (Delphi)/GUI class (Java), as in the section that follows.

NOTE: The object **objCardHolder** has been declared globally as part of the provided code.

2.2.1 **Button – [2.2.1 – Check access code]**

The user is required to select a card number from the four card numbers provided in the combo box.

Once a card number has been selected in the combo box, the cellphone number of the card holder and the number of loyalty points already earned from previous purchases are displayed with the provided code.

The user must enter the correct access code for the selected card number to allow the program to process the data of the card holder saved in the text file.

Write code to do the following:

- Use the card number, cellphone number and loyalty points of the card holder to instantiate a card holder object.

- Use the method **isCorrect** to determine whether the access code that was entered is the correct access code for this card holder or not.

If the correct access code has been entered, use the data in the **DataJanuary2017.txt** text file to determine the following regarding the purchases of the card holder:

- The total amount spent for the month
- The amount spent on health-related products
- The number of times the user visited the shop and made purchases

Use this information and the relevant methods from the object class to update the number of loyalty points, the health level code and the number of times the card holder visited the shop for purchases. Enable the button for QUESTION 2.2.2.

If the access code is NOT correct, display a message to indicate that the login was not successful and clear the access code text box.

The data in the table below can be used for testing purposes.

Card number	Access code
B921-3$-DI	188
B870-4%-CH	41
P730-7*-DI	338
B810-2#-DI	170

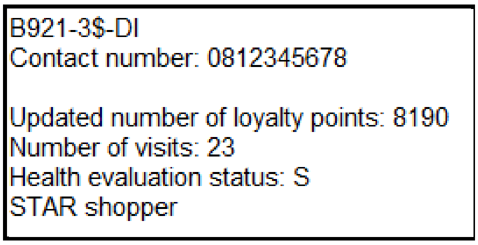
(24)

2.2.2 **Button – [2.2.2 – Display card holder details]**

Write code to use the **toString** method to display the card holder object data in the output area provided.

The method created in QUESTION 2.1.6 must be used to indicate if the card holder is a STAR shopper.

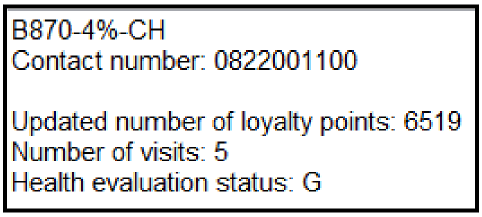
Example of output for card number B921-3$-DI:



```
B921-3$-DI
Contact number: 0812345678

Updated number of loyalty points: 8190
Number of visits: 23
Health evaluation status: S
STAR shopper
```

Example of output for card number B870-4%-CH:



```
B870-4%-CH
Contact number: 0822001100

Updated number of loyalty points: 6519
Number of visits: 5
Health evaluation status: G
```

(3)

- Enter your examination number as a comment in the first line of the class and the form.
- Save all the files.
- Print the code contained in the class and the form (Delphi)/both classes (Java) if required.

TOTAL SECTION B: 65

SECTION C

QUESTION 3: PROBLEM-SOLVING PROGRAMMING

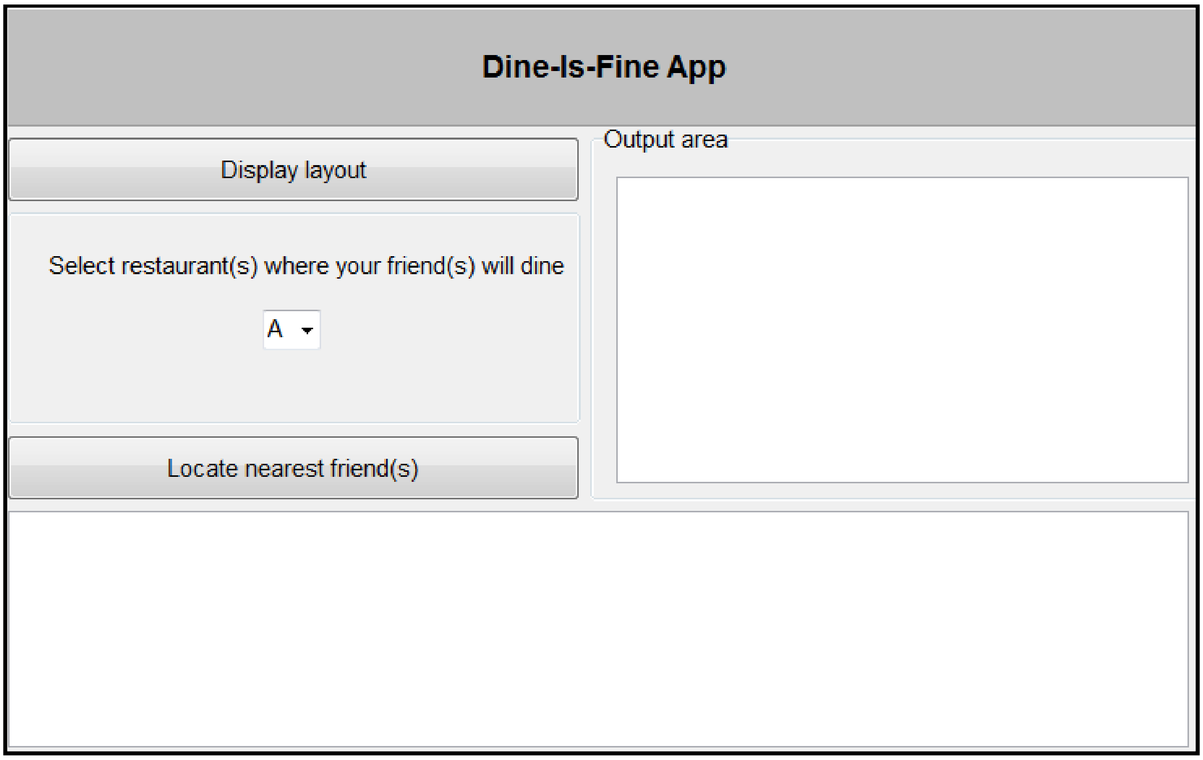
SCENARIO

A social media companion app called *Dine-Is-Fine* has been designed specifically for the 10 restaurants at your local mall. You and your friends are registered users of this app. The app can be used to share your location with other registered users.

Do the following:

- Compile and execute the program in the **Question3** folder. Currently the program has no functionality.
- Complete the code for each question, as described in QUESTION 3.1 to QUESTION 3.3.

Supplied GUI:



Supplied data:

You are provided with a two-dimensional array called **arrMall** filled with characters that represent the layout of the mall. The characters A to J represent 10 restaurants that you and your friends regularly visit.

Output area:

Decide on a suitable output component to be placed in the output area provided to display the content of the two-dimensional array.

NOTE:

- You are NOT allowed to modify supplied data manually. Code must be written to manipulate the supplied data according to the requirements.

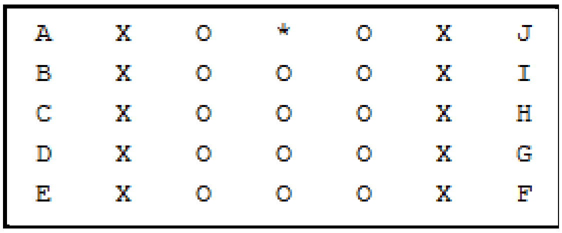
- Good programming techniques and modular design must be applied in the design and coding of your solution.

- **NO marks will be assigned for hardcoding**. Use control structures and variables where necessary.

3.1 **Button [Display layout]**

The program must display the layout of the mall. Write code to use the data supplied in the two-dimensional array to display the layout in neat columns.

Example of output:



```
A    X    O    *    O    X    J
B    X    O    O    O    X    I
C    X    O    O    O    X    H
D    X    O    O    O    X    G
E    X    O    O    O    X    F
```

(5)

Explanation of the data in the array:

- The characters A to J represent the position of each of the 10 restaurants in the mall.
- The character X is a placeholder indicating that none of your friends are currently in a restaurant.
- The character O indicates the open area where one can move around.
- The asterisk character (*) indicates your position (first row and fourth column).

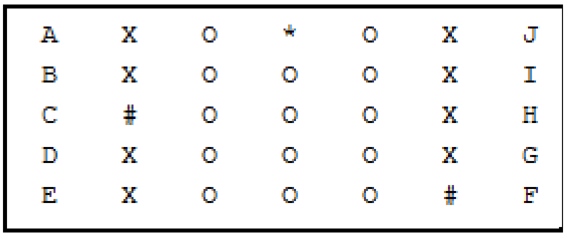
NOTE: You will always remain in the same position for the purpose of this question.

3.2 **Combo box with restaurant codes**

You are requested to select a restaurant from the combo box where one of your friends will dine.

Write code to replace the X placeholder at the selected restaurant with a hashtag symbol (#) to indicate that a friend is at the restaurant and display an updated layout of the mall.

Example of output if two of your friends are dining at restaurant C and F respectively:



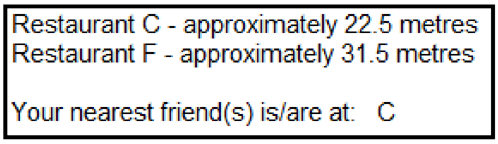
```
A   X   O   *   O   X   J
B   X   O   O   O   X   I
C   #   O   O   O   X   H
D   X   O   O   O   X   G
E   X   O   O   O   #   F
```

(13)

3.3 **Button [Locate nearest friend(s)]**

The program must determine at which restaurant(s) in the mall your friends currently are and how far they are from where you are. Every vertical or horizontal movement on the matrix represents a distance of 4,5 metres. From the restaurant(s) where your friends are, determine and display the restaurant(s) closest to where you are.

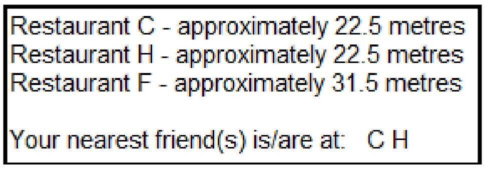
Example of output if your friends are dining at restaurants C and F:



```
Restaurant C - approximately 22.5 metres
Restaurant F - approximately 31.5 metres

Your nearest friend(s) is/are at:   C
```

Example of output if your friends are dining at restaurants C, F and H:



```
Restaurant C - approximately 22.5 metres
Restaurant H - approximately 22.5 metres
Restaurant F - approximately 31.5 metres

Your nearest friend(s) is/are at:   C H
```

NOTE: You cannot move diagonally on the matrix.

MARK ALLOCATION FOR QUESTION 3.3

Locate the friends	3
Determine the name of the restaurant	4
Calculate the distance	6
Find the nearest friend(s)	3
Output to indicate position of friend(s) and distance	2
Correct format of output for the nearest friend(s)	1
TOTAL	**19**

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

TOTAL SECTION C: 37
GRAND TOTAL: 150