

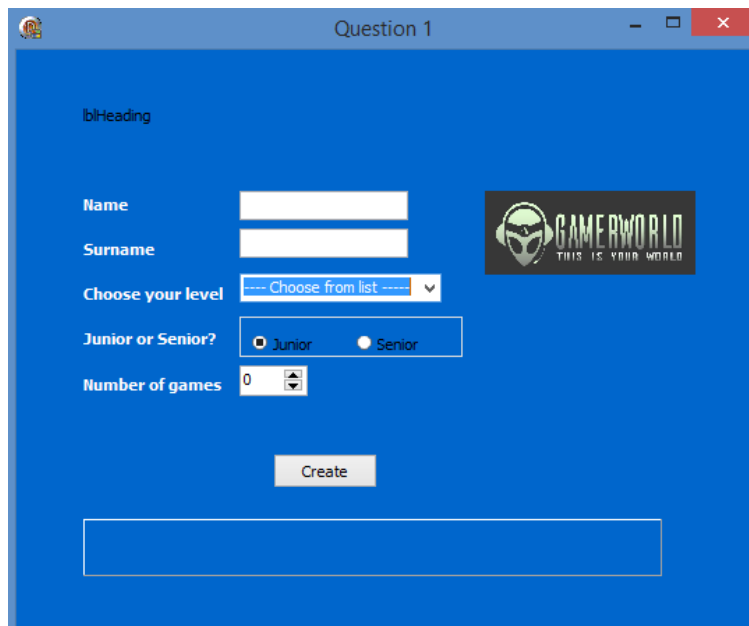
| | | |
|---------------------|---|--|
| TERM 4 | Week 4 | |
| TOPIC | Software Development – Problem Solving | |
| AIMS OF LESSON | Develop problem solving skills | |
| RESOURCES | Paper based resources | Digital resources |
| | DBE Practical Textbook Gr 10 (Use school issued textbook for the same content) | Links on the WCED ePortal: DataFiles The e-resources below has the actual URL to the websites |
| Introduction | Reinforce problem solving principals | |
| CONCEPTS AND SKILLS | Problem solving | |

QUESTION 1: GENERAL PROGRAMMING SKILLS

- **Open the project Question1.**
- **Open the incomplete class called frmQuestion1_u.**
- **Complete the code for each section of QUESTION 1 as described in QUESTION 1.1 to QUESTION 1.6 below**

Gamer World is a small enterprise offering gaming fanatics the chance to play and compete against each other for the day. Gamers need a unique client code which they have to use when Registering to play games. This unique code is created by using the client’s name, surname, level he/she is playing at and a J or a S depending on whether he/she is a junior or a senior at school.

The following GUI is displayed:



Button btnCreate [Create]

- 1.1. Change the design of the GUI
In the object inspector or the use of code, change the heading to “Gamers World Client Information”, set the font to a white colour and the size of 20 pts. It should also be displayed in bold. (4)
- 1.2. Declare a constant for the base cost to register, which is R100. (1)
- 1.3. Declare variables needed for storing the data entered by the user – name, surname, level (Beginner, Intermediate or Advanced), indication of junior or senior (store ‘J’ or ‘S’) and number of games. Use meaningful names and Delphi convention taught. (6)

1.4. Input

Read the data from the components and store it in the data structures (variables) that you have created:

- the name of the gamer from the *TEdit*, **edtName**,
- the surname of the gamer from *TEdit* **edSurname**,
- the level he/she is playing from *TComboBox*, **cmbLevel**,
- whether he/she is a junior or a senior at school from the *TRadioGroup*, **rgpJuniorSenior**, Store only the first character.
- the number of games the gamer wishes to play from the *TSpinEdit*, **sedNumGames**

(7)

1.5. Processing:

Create the code for the gamer by concatenating the first character of the name, the first character of the surname, the first character of the level and the 'J'/'S' to indicate junior or senior.

Determine the cost for the gamer. The base cost to register is R100 if you'll be playing 3 or less games (use the constant declared). Each game thereafter will add R10 to the cost.

(6)

1.6. Output:

Display the client code together with a message, ' – cost is R', and the value of the cost calculated.

(6)

Examples:



Total Question 1: [30]

QUESTION 2: PROGRAMMING APPLICATIONS

- **Open the project Question2.**
- **Open the incomplete class called frmQuestion2_u.**
- **Complete the code as describe in QUESTION 2.1 to QUESTION 2.3 below.**

People are creatures of habit. We tend to use the same PIN number again and again. This is not a safe habit. Your task is to write a program that can generate PIN numbers of different lengths.

- 2.1. In the Question 2.1 groupbox variable length pin numbers can be created by typing a number in the Input edit field and clicking on the button. This will add the number to the end of the number displayed in the Pin label. It is not necessary to validate the input from the user. The user will always enter a positive single digit number. Only test for **0**.

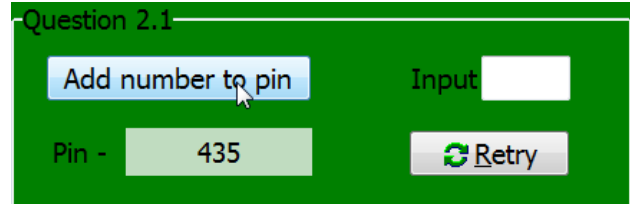
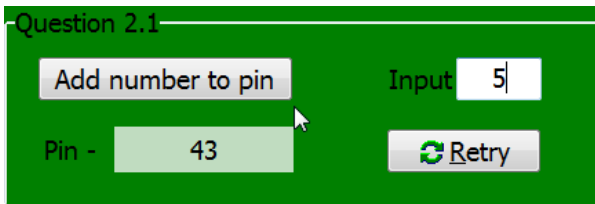
2.1.1. **Button btnCreate [Add number to PIN]**

Create a global text type variable in which to store your pin number as it is generated. The user must enter a single digit number, excluding **0**, in the edit box **edtInput**. If the user enters 0 it must not be added to the pin number. When the button is clicked, the number must be added to the pin and displayed in the label, **lblIPin** (next to **Pin -**).

Also clear the input field.

Example:

User enters 4. After first click: Pin – 4
 User enters 0. After second click: Pin – 4
 User enters 3 After third click: Pin – 43
 User enters 5 After fourth click: Pin – 435
 and so on



Note: Do not use a loop structure to solve the problem. (9)

2.1.2 **Button btnClear [Retry]**

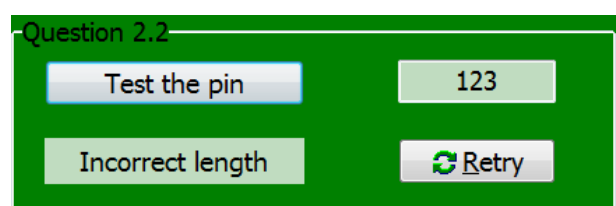
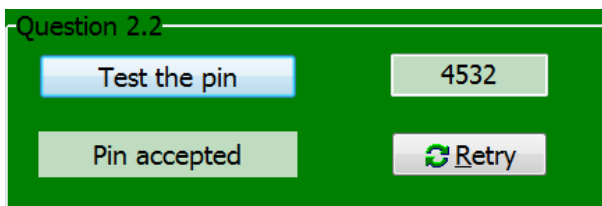
When the button is clicked, the Pin label, input field and the global variable must be cleared. (3)

2.2. Some users want to use their own pin numbers. It is your task to test that their pin numbers are legal.

2.2.1. **Button btnTestPin [Test the pin]**

Test to see that the length of the pin is 4 or more and less than 7 digits. Display a message in the empty label provided, **lblTest** that will tell the user that the Pin is accepted or that the length is incorrect. Display the “Pin accepted” message in green and the “Incorrect length” message in red.

Examples:



(10)

2.2.2. **Button btnRetry2 [Retry]**

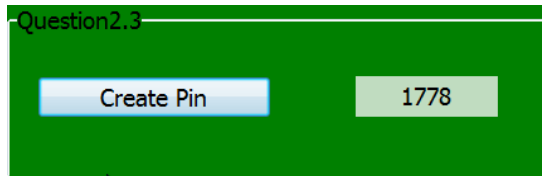
Write code to clear the input, **edtPin** and the message label, **lblTest**.

(2)

2.3. **Button btnCreatePin [Create Pin]**

Most Pin numbers used with banking cards are 4 digits long. Write code to generate a random 4 digit pin number from 1000 to 9999 (both included). Display this number in the provided label, **lblRandomPin**.

Example:



The value of the random pin will be different from the example given, each time you run the program.

(6)

Total Question 2: [30]

QUESTION 3: PROBLEM SOLVING

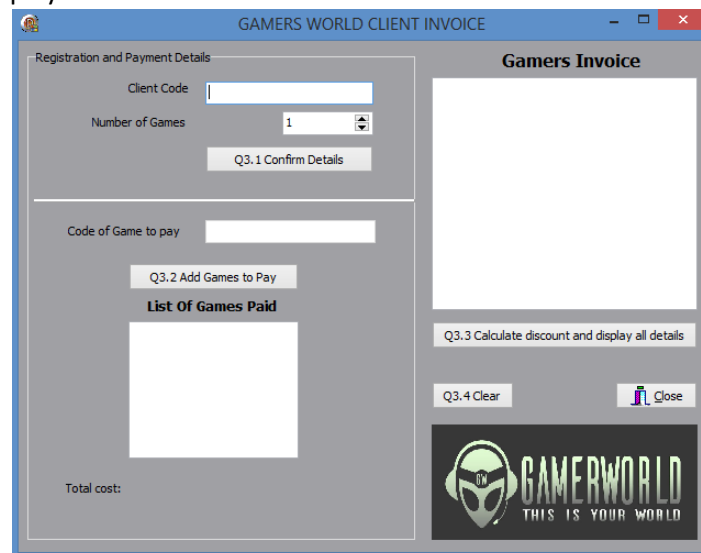
- Open the project Question3.
- Open the incomplete class called frmQuestion3_u.
- Complete the code as describe in QUESTION 3.1 to QUESTION 3.4 below.

Gamer World restricts the number of games that individuals may register to play for the day. Currently clients may only register for up to 4 games. Additionally, to encourage individuals to play a variety of games, different types of discounts are awarded based on the total cost.

The games with their fees for the day are shown below:

| Games Codes | Games Fee |
|-------------|-----------|
| 2016 | R60.00 |
| 2017 | R70.00 |
| 2018 | R80.00 |
| 2019 | R100.00 |

The following GUI is displayed:

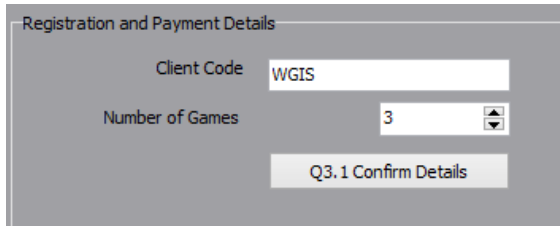


3.1. Button btnConfirm [Q3.1 Confirm Details]

Obtain the number of games from the spin edit, **sedNumber**.

Current clients are only allowed to register for **1 to 4** games. Test whether the correct number has been entered. If so, display **'Games Confirmed'** in a message box, and make the "Code of Game to play", **edtGameCode** component **editable**. If not, display the client code entered in the **TEdit**, **edtClientCode**, with a message **' - please select again'** in a message box, and disable the **edtGameCode** component.

Examples:

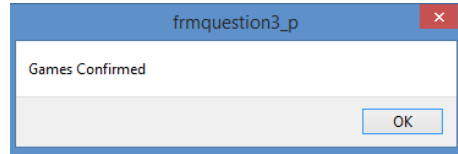


Registration and Payment Details

Client Code: WGIS

Number of Games: 3

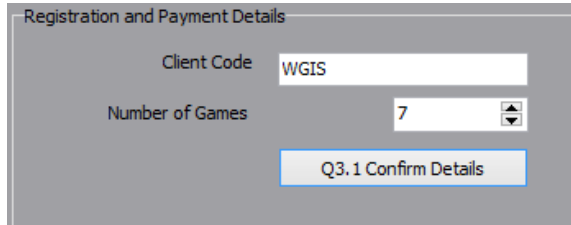
Q3.1 Confirm Details



frmquestion3_p

Games Confirmed

OK

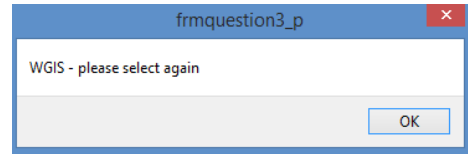


Registration and Payment Details

Client Code: WGIS

Number of Games: 7

Q3.1 Confirm Details



frmquestion3_p

WGIS - please select again

OK

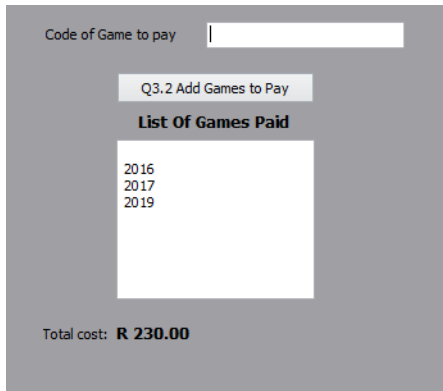
(10)

3.2. Button btnAdd [Q3.2 Add Games To Pay]

The user must enter the code for the game he/she wishes to pay for in the *TEdit*, **edtGameCode** .
 Validate the code entered. Valid codes were given in the introduction above.
 If the code is correct, add the code to *TMemo*, **memPaid** and add the **cost of the game** to the **rCost** (globally declared already).
 If the game code is incorrect, display in a message box '**Wrong Game Code. Please enter a correct code.**'.
 In both cases clear the *TEdit*, **edtGameCode** and set the focus to this component.

Examples:

The user has entered 3 game codes, 2016, 2017 and 2019:



Code of Game to pay

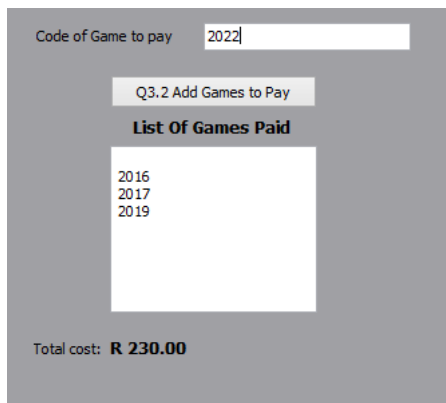
Q3.2 Add Games to Pay

List Of Games Paid

2016
2017
2019

Total cost: **R 230.00**

The user has entered an incorrect code:



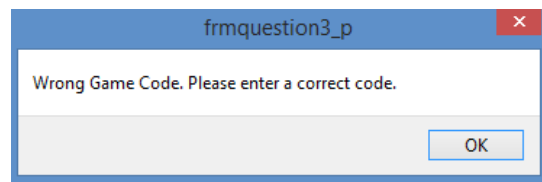
Code of Game to pay: 2022

Q3.2 Add Games to Pay

List Of Games Paid

2016
2017
2019

Total cost: **R 230.00**



frmquestion3_p

Wrong Game Code. Please enter a correct code.

OK

(16)

INFORMATIONTECHNOLOGY: GRADE 10

3.3. Button btnCalculate [Q3.3 Calculate discount and display details]

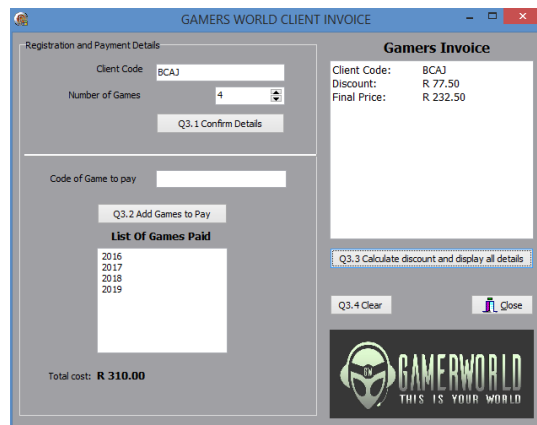
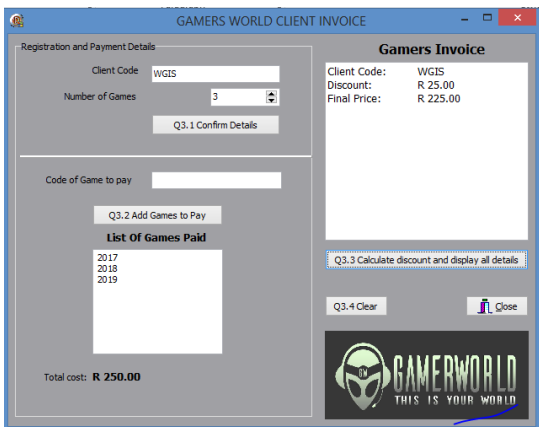

Read the Client Code from the edit box, `edtClientCode` and display it in the *TRichEdit*, **redInvoice**. Calculate and display the discount awarded. The discount is awarded based on the following criteria:

- A super discount rate of 25% if the total cost is greater than or equal to R300.
- A mid discount rate of 10% if the total cost is greater than R200.
- A low discount rate of 5% if the total cost is greater than or equal to R150.

Calculate and display the final price after the discount has been subtracted.

Display these values with suitable captions and ensure that the above 3 line are neatly displayed in 2 columns.

Examples:

(10)

3.4. Button btnClear [Q3.4 Clear]

Set the global variable `rTotalCost` to 0. Clear the total cost label, all edit boxes and output areas. (2)

Total Question 3: [40]

TOTAL: 100

| | |
|---------------|--|
| CONSOLIDATION | <i>In this problem you used all your skills to complete the problems.</i> |
| VALUES | <i>Thank you for working through this document! Stick to your hard work. You are creating an opportunity to better yourself. Your perseverance will bring dividends.</i> |
| E-RESOURCES | Data files: https://wcedportal.co.za/eresource/169526 |