Object-Oriented Programming (DSE 2123) Mini Project Submission

Student Name:	Keerthan Kumar C
Reg No	220968002
Assignment No	FISAC
Subject Code	DSE 2123
Subject	OOP with Java
Marks	10M

Overview

Problem Statement:

The objective of this project is to develop a graphical user interface (GUI) application in Java that facilitates the administration of an admission entrance test. This test comprises a series of Multiple-Choice Questions (MCQs) and is structured as follows:

1. ***Login Window:***

- The application initiates by presenting a login window, designed for user authentication.

- Users must input their credentials, including a username and password, for validation purposes.

- Upon successful authentication, the application transitions to a new page.

2. ***MCQ Page:***

- On the MCQ page, the application presents a set of 5 MCQs, each featuring a question prompt and a set of radio buttons for selecting answer choices.

- Users can select a single answer choice per question.

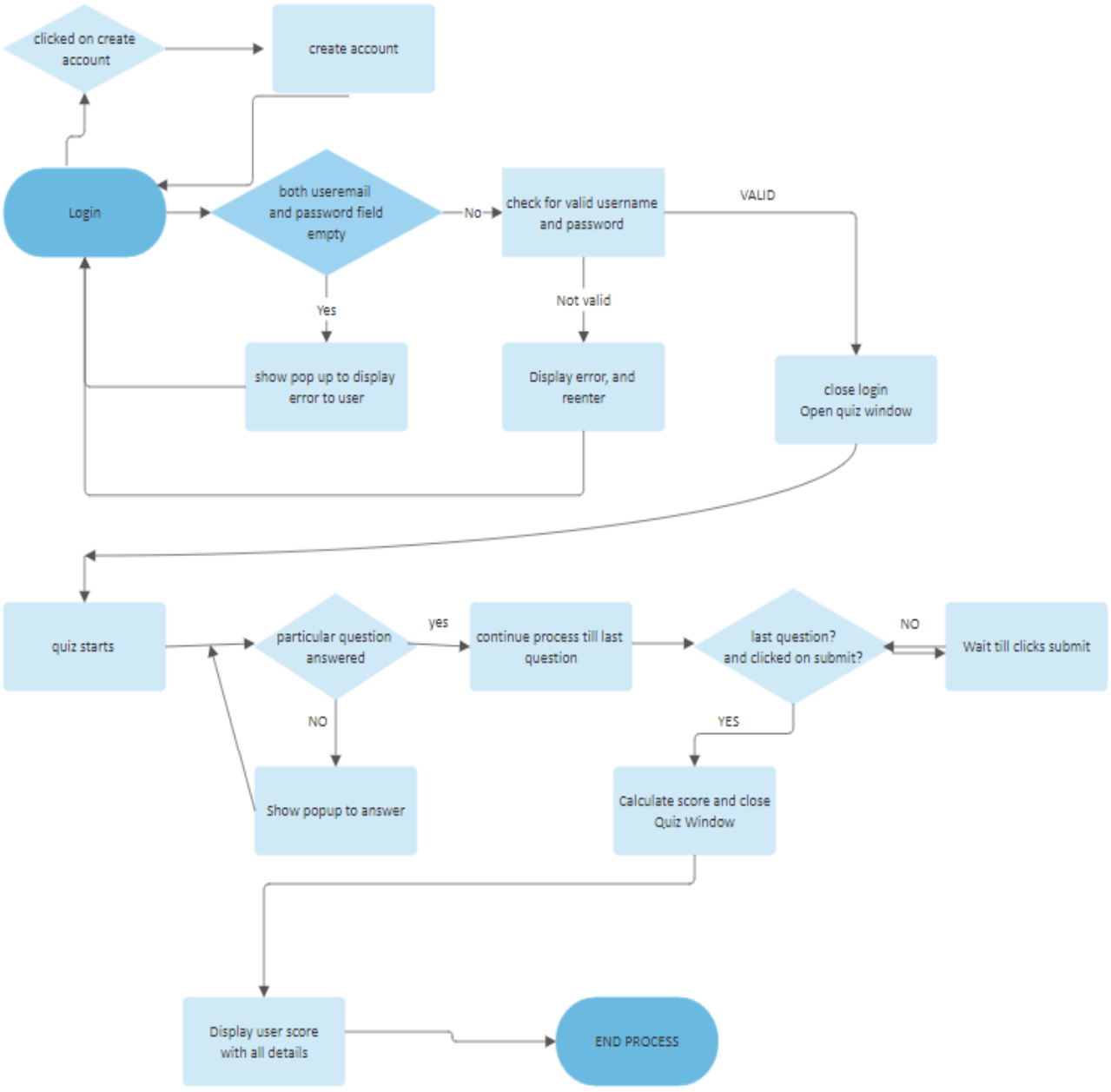
- A "Submit" button is provided to allow users to submit their responses.

3. ***Result Display:***

- Once users have responded to all the questions and submitted their answers, the application calculates and exhibits the total score attained by the user.

This project aims to offer a comprehensive solution for conducting an admission entrance test, providing an efficient and user-friendly interface for both test-takers and administrators.

Flow diagram



Components Used in LoginForm Class:

1. **`JFrame`**: Represents the main frame of the application.

2. **`JPanel`** (left and right): Panels used for layout and design.

3. **`JLabel`** (User, Pass, Message, logo, wel, login, n, createAccountLabel): Labels for displaying text, icons, and messages.

4. **`JButton`** (Login, Reset): Buttons for user interaction.

5. **`JTextField`** (username): Input field for the username.

6. **`JPasswordField`** (password): Input field for the password.

7. **`JToggleButton`** (showPassword): Toggle button for showing/hiding the password.

8. **`ImageIcon`**: Used to load and display images.

9. **`Cursor`**: To set the cursor style for the "Create Account" label.

Events and Actions Used in LoginForm Class:

1. **`actionPerformed(ActionEvent e)`**: This method handles actions when buttons are clicked.

   - If the "Reset" button is clicked, it clears the username and password fields and the error message.

   - If the "Login" button is clicked, it:

     - Validate the input by checking if the username and password are in the correct format.

     - If the input is not valid, it displays an error message.

     - If the input is valid, it disposes of the login window and opens a new quiz window with the provided username.

2. **`itemStateChanged(ItemEvent e)`**: This method handles the state change of the "Show" toggle button.

   - If the "Show" button is selected (toggled), it sets the password field to show the password characters.

   - If the "Show" button is deselected, it sets the password field to mask the password characters.

The LoginForm Class uses various components and event handling to create a functional login window for the project. The `isUsernameValid` and `isPasswordValid` methods validate the format of the username and password, respectively, ensuring they meet certain criteria.

This LoginForm Class sets the foundation for user authentication before proceeding to the quiz part of your application.

Components Used in QuizWindow Class:

1. **`JFrame`**: Represents the main frame of the application.
2. **`JPanel`** (rules, Header, main): Panels used for layout and design.
3. **`JLabel`** (RUles, Userinfo, label, question): Labels for displaying text and images.
4. **`JButton`** (next, submit): Buttons for user interaction.
5. **`JRadioButton`** (opt1, opt2, opt3, opt4): Radio buttons for selecting answer options.
6. **`ButtonGroup`**: Groups radio buttons to ensure exclusive selection.
7. **`ImageIcon`**: Used to load and display images.
8. **`JOptionPane`**: Displays dialog boxes for user prompts and messages.

Events and Actions Used in QuizWindow Class:

1. **`actionPerformed(ActionEvent e)`**: This method handles actions when buttons are clicked.

   - If the "NEXT" button is clicked, it checks if an option is selected. If yes, it asks for confirmation to move to the next question.

   - If the user confirms and selects an option, it increments the count and loads the next question.

   - If the "SUBMIT" button is clicked, it checks if an option is selected. If yes, it asks for confirmation to submit the quiz.

   - If the user confirms and selects an option, it calculates the score and opens a new window with the score.

Methods:

1. **`start(int count)`**: Sets up the question and answer options based on the current count.

The quiz window of this Java GUI application manages the display of questions, options, and user interactions, as well as handles the scoring and submission of the quiz.

Components Used in Score Class:

1. **`JFrame`**: Represents the main frame of the application.
2. **`JLabel`** (label, Userinfo): Labels for displaying text and images.
3. **`JButton`** (exit): Button for user interaction.
4. **`ImageIcon`**: Used to load and display images.

Events and Actions Used in Score Class:

1. **`actionPerformed(ActionEvent e)`**: This method handles actions when buttons are clicked.

   - If the "Exit" button is clicked, it disposes of the current frame, effectively closing the application.

This Score Class represents the score window of your Java GUI application. It displays the user's details and score after completing the quiz. The "Exit" button allows the user to close the result window. The details displayed include the user's name, college, registration number, department, and the obtained score out of 50.

Program Code:

1. LoginForm Class:

https://drive.google.com/file/d/16jTOZB1Ag9rDOFOj-DO7UEEf85lI0f21/view?usp=sharing

```
1
2  package javaprojectfiles;
3  //Importing all necessary packages
4  import java.awt.Color;
5  import java.awt.Cursor;
6  import java.awt.Desktop;
7  import java.awt.Font;
8  import java.awt.event.ActionEvent;
9  import java.awt.event.ActionListener;
10 import java.awt.event.ItemEvent;
11 import java.awt.event.ItemListener;
12 import java.awt.event.MouseAdapter;
13 import java.awt.event.MouseEvent;
14 import java.io.IOException;
15 import java.net.URI;
16 import java.net.URISyntaxException;
17 import java.util.regex.Pattern;
18 import javax.swing.BorderFactory;
19 import javax.swing.ImageIcon;
20 import javax.swing.JButton;
21 import javax.swing.JFrame;
22 import javax.swing.JLabel;
23 import javax.swing.JPanel;
24 import javax.swing.JPasswordField;
25 import javax.swing.JTextField;
26 import javax.swing.JToggleButton;
27 import javax.swing.border.LineBorder;
28 //Creating a login form class by extending JFrame
29 public class LoginForm extends JFrame implements ActionListener, ItemListener{
30     //Declaring all necessary Components required
31     JPanel left;JLabel User;JLabel Pass;
32     JPanel right;JLabel Message;JButton Login,Reset;
33     JLabel logo;JLabel wel;JLabel login;JLabel n;
34     JTextField username;JPasswordField password;
35     JToggleButton showPassword;
36     JLabel createAccountLabel = new JLabel("Dont have an account?Create");
```

```
37      //Constructor of the loginform class
38      LoginForm()
39      {
40          //Creating image icon
41          ImageIcon icon=new ImageIcon("download.jpg");
42          ImageIcon user=new ImageIcon("username.png");
43          ImageIcon pass=new ImageIcon("password.png");
44
45          //new panels left and right
46          left=new JPanel();
47          left.setBackground(new Color(255,255,255));
48          left.setBounds(0, 0, 450, 500);
49          this.add(left);
50          right=new JPanel();
51          right.setBackground(new Color(255,128,0));
52          right.setBounds(450, 0, 550, 500);
53          this.add(right);
54          left.setLayout(null);
55          right.setLayout(null);
56
57          //Adding logo on the left panel
58          logo=new JLabel();
59          logo.setForeground(new Color(255,128,0));
60          logo.setBounds(110,100,icon.getIconWidth(),icon.getIconHeight());
61          logo.setIcon(icon);
62          left.add(logo);
63
64          //new labels on left
65          wel=new JLabel();
66          wel.setText("WELCOME TO MANIPAL ACADEMY OF HIGHER EDUCATION");
67          wel.setBounds(30,30,400,50);
68          wel.setFont(new Font("Peterdraw",Font.BOLD,13));
69          left.add(wel);
70
```

```
70
71          login=new JLabel();
72          login.setText("Login with MAHE Microsoft Account");
73          login.setBounds(45,350,400,50);
74          login.setFont(new Font("Peterdraw",Font.BOLD,20));
75          left.add(login);
76
77          //new components on right
78          n=new JLabel();
79          n.setText("LOGIN");
80          n.setBounds(210,40,150,50);
81          n.setFont(new Font("ILOTT-TYPE",Font.BOLD,40));
82          n.setBackground(Color.WHITE);
83          n.setForeground(Color.WHITE);
84          right.add(n);
85
86          User=new JLabel();
87          User.setIcon(user);
88          User.setBounds(70,110,user.getIconWidth(),user.getIconHeight());
89          right.add(User);
90
91          Pass=new JLabel();
92          Pass.setIcon(pass);
93          Pass.setBounds(70,190,pass.getIconWidth(),pass.getIconHeight());
94          right.add(Pass);
95
96          username=new JTextField();//username textfield
97          username.setBounds(120, 120, 300, 40);
98          username.setBackground(Color.white);
99          username.setFont(new Font("ILOTT-TYPE",Font.PLAIN,20));
100         right.add(username);
101
102         password=new JPasswordField();//password field
103         password.setBounds(120, 200, 300, 40);
104         password.setBackground(Color.white);
105         password.setEchoChar('*');
106         password.setFont(new Font(null,Font.PLAIN,20));
107         right.add(password);
108
```

```
108
109         Message=new JLabel();//show error message
110         Message.setBounds(140,245,350,20);
111         Message.setForeground(Color.RED);
112         Message.setFont(new Font(null,Font.ITALIC,15));
113         right.add(Message);
114
115         Login=new JButton("Login");//login button
116         Login.setBounds(130,300,120,40);
117         Login.setFocusable(false);
118         Login.addActionListener(this);
119         Login.setBackground(Color.white);
120         Login.setFont(new Font(null,Font.ITALIC,15));
121         Login.setBorder(BorderFactory.createEtchedBorder());
122         right.add(Login);
123
124         Reset=new JButton("Reset");//reset button
125         Reset.setBounds(300,300,120,40);
126         Reset.setFocusable(false);
127         Reset.addActionListener(this);
128         Reset.setBackground(Color.white);
129         Reset.setFont(new Font(null,Font.ITALIC,15));
130         Reset.setBorder(BorderFactory.createEtchedBorder());
131         right.add(Reset);
132
133         showPassword = new JToggleButton("Show");//a toggle button to show password
134         showPassword.setBounds(440, 210, 70, 30);
135         showPassword.setFocusable(false);
136         showPassword.addItemListener(this);
137         showPassword.setBackground(Color.white);
138         showPassword.setFont(new Font(null,Font.ITALIC,15));
139         showPassword.setBorder(BorderFactory.createEtchedBorder());
140         right.add(showPassword);
141
141
142         createAccountLabel.setBounds(180, 380, 350, 30);//label to direct to other link
143         createAccountLabel.setForeground(Color.WHITE);
144         createAccountLabel.setFont(new Font("Peterdraw",Font.ITALIC,15));
145         createAccountLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
146         createAccountLabel.addMouseListener(new MouseAdapter() {
147             // Adding mouse listener to the createAccountLabel
148             public void mouseClicked(MouseEvent e) {
149                 //Implementing the mouseClicked method for the MouseListener
150                 try {
151
152                     Desktop.getDesktop().browse(new URI("https://signup.live.com/"));
153                 } catch (IOException  ex) {
154                     ex.printStackTrace();
155                 } catch (URISyntaxException e1) {
156                     e1.printStackTrace();
157                 }
158             }
159         });
160         right.add(createAccountLabel);
161
162         //Frame Properties defining the frame
163         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
164         this.setLayout(null);
165         this.setBounds(200, 100, 1000, 500);
166         this.setTitle("LOGIN-Manipal Academy of Higher Education");
167         this.setIconImage(icon.getImage());
168         this.setBackground(Color.white);
169         this.setResizable(false);
170         this.setVisible(true);
171     }
```

```
172     //Implementing the actionPerformed method
173     public void actionPerformed(ActionEvent e) {
174         if(e.getSource()==Reset)//if the user clicks on reset
175         {
176             username.setText("");
177             password.setText("");
178             Message.setText("");
179         }
180         if(e.getSource()==Login)//if user clicks on login
181         {
182             String log=username.getText();
183             String pas=String.valueOf(password);
184             if(log.isEmpty() || pas.isEmpty())//login clicked but fields are empty
185             {
186                 Message.setText("Please enter a Valid Input");
187             }
188             else
189             {   //if all fields are not empty checking validaton and giving appropriate error messages
190                 if (!isUsernameValid(log) && !isPasswordValid(pas)) {
191                     Message.setText("Invalid username and password format.");
192                 } else if (!isUsernameValid(log)) {
193                     Message.setText("Invalid username format.");
194                 } else if(!isPasswordValid(pas)){
195                     Message.setText("Invalid password format.");
196                 }
197                 else
198                 {   //if everything is correct opening quiz window
199                     this.dispose();
200                     new quizWindow(log);
201                 }
202             }
203         }
204 
205     }
206     private boolean isUsernameValid(String username) {
207         // Username should contain "@" and end with ".com" and be all in lowercase
208         String regex = "^[a-z0-9]+@.*\\.com$";
209         return Pattern.matches(regex, username);
210     }
211 
212     //Helper method to check if the password is in the correct format
213     private boolean isPasswordValid(String password) {
214         // Password should contain at least one uppercase letter and one digit
215         return Pattern.compile("^(?=.*[A-Z])(?=.*\\d).+$").matcher(password).matches();
216     }
217 
218     public void itemStateChanged(ItemEvent e) {
219         if (e.getStateChange() == ItemEvent.SELECTED) {
220             password.setEchoChar((char) 0); // Show the password as a string (no masking)
221         } else {
222             password.setEchoChar('*'); // Mask the password
223         }
224     }
225 
226 }
227 
```

2. QuizWindow Class:

https://drive.google.com/file/d/1sEE2zdVeEDCEMtaEn-CpIIgvaBampGdh/view?usp=sharing

```java
package javaprojectfiles;
//Importing all necessary packages
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.BorderFactory;
import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
import javax.swing.JRadioButton;

//Creating a quizwindow class by extending JFrame
public class quizWindow extends JFrame implements ActionListener{
    //Declaring all necessary Components required
    ImageIcon icons,logo;
    JRadioButton opt1,opt2,opt3,opt4;
    ButtonGroup group;
    JLabel label,question,RUles,Userinfo;
    JButton next,submit;
    JPanel rules,Header,main;

    //Storing the questions and options in a 2d array of strings
    String questions[][]=new String[5][5];{
    questions[0][0] = "1.Which is used to find and fix bugs in the Java programs.?";
    questions[0][1] = "JVM";
    questions[0][2] = "JDB";
    questions[0][3] = "JDK";
    questions[0][4] = "JRE";
```

```
35
36      questions[1][0] = "2.What is the return type of the hashCode() method in the Object class?";
37      questions[1][1] = "int";
38      questions[1][2] = "Object";
39      questions[1][3] = "long";
40      questions[1][4] = "void";
41
42      questions[2][0] = "3.Which package contains the Random class?";
43      questions[2][1] = "java.util package";
44      questions[2][2] = "java.lang package";
45      questions[2][3] = "java.awt package";
46      questions[2][4] = "java.io package";
47
48      questions[3][0] = "4.An interface with no fields or methods is known as?";
49      questions[3][1] = "Runnable Interface";
50      questions[3][2] = "Abstract Interface";
51      questions[3][3] = "Marker Interface";
52      questions[3][4] = "CharSequence Interface";
53
54      questions[4][0] = "5.In which memory a String is stored, when we create a string using new operator?";
55      questions[4][1] = "Stack";
56      questions[4][2] = "String memory";
57      questions[4][3] = "Random storage space";
58      questions[4][4] = "Heap memory";
59      }
60      //Storing the Answers in a 2d array of strings
61      String answers[][]=new String[5][2];{
62      answers[0][1] = "JDB";
63      answers[1][1] = "int";
64      answers[2][1] = "java.util package";
65      answers[3][1] = "Marker Interface";
66      answers[4][1] = "Heap memory";}
67
68      //Creating a 2d array to store the user's answers
69      String useranswers[][] = new String[5][1];
70      static int count=0;
71      int score=0;String username;
72
```

```
73      //Constructor of the quizwindow class
74      quizWindow(String name)
75      {
76          //Extracting the username from the name passed from loginform class
77          username=name;
78          int ind=username.indexOf('@');
79          username=username.substring(0,ind);
80
81          //getting images and setting logo
82          ImageIcon icons=new ImageIcon("download.jpg");
83          ImageIcon logo=new ImageIcon("m.jpg");
84          label=new JLabel(logo);
85          label.setBounds(90, 0,logo.getIconWidth(),logo.getIconHeight());
86
87          //Creating a header panel to dd logo to
88          Header=new JPanel();
89          Header.setBounds(0,0,1000,logo.getIconHeight());
90          Header.setBackground(Color.orange);
91          Header.add(label);
92
93          //question labe lto display question
94          question=new JLabel();
95          question.setBounds(150,200,700,100);
96          question.setFont(new Font("Peterdraw",Font.BOLD,16));
97
98          //Main panel to have questions,options,rules,username and exit
99          main=new JPanel();
100         main.setBounds(100,logo.getIconHeight()+10,logo.getIconWidth() ,400);
101         main.setBackground(Color.white);
102
103         //A panel for rules and label for the rules to display the rules to the user
104         rules=new JPanel();
105         rules.setBounds(600, 280, 350, 350);
106         rules.setBackground(new Color(255,255,255));
107         rules.setBorder(BorderFactory.createEtchedBorder());
108         rules.setLayout(null);
109         rules.setFont(new Font("Tahoma", Font.PLAIN, 16));
110         RUles=new JLabel();
111         RUles.setBounds(20, 5, 700, 350);
```

```
111         RUles.setBounds(20, 5, 700, 350);
112         RUles.setFont(new Font("Tahoma", Font.PLAIN, 14));
113         RUles.setText(
114             "<html>"+
115                 "READ THE RULES BEFORE STARTING THE QUIZ!!" + "<br><br>" +
116                 "QUIZ RULES" + "<br><br>" +
117                 "1.All questions are compulsory." + "<br><br>" +
118                 "2.Each question carries 10 points." + "<br><br>" +
119                 "3.There is no time limit for the quiz." + "<br><br>" +
120                 "4.You are not allowed to use any resources." + "<br><br>" +
121                 "5.You are not allowed to communicate with anyone." + "<br><br>" +
122                 "6.Any attempt to cheat will result in disqualification." + "<br><br>"+
123 
124             "<html>"
125         );
126         rules.add(RUles);
127 
128         //Showing the username on the screen
129         Userinfo=new JLabel(username);
130         Userinfo.setFont(new Font("Tahoma", Font.PLAIN, 13));
131         Userinfo.setBounds(785, 190, 190, 80);
132         Userinfo.setBorder(BorderFactory.createLineBorder(Color.black));
133         Userinfo.setText(
134             "<html>"+
135                 "Student Detials" + "<br>" +
136                 "Name: "+username + "<br>" +
137                 "Registration Number: 220968002" + "<br>" +
138              "<html>"
139                 );
140 
141         //Creating radiobuttons
142         opt1=new JRadioButton("Option 1");
143         opt1.setBounds(150,300,200,30);
144         opt1.setFocusable(false);
145         opt1.setBackground(Color.white);
146         this.add(opt1);
147 
```

```
147 
148         opt2=new JRadioButton("Option 2");
149         opt2.setBounds(150,350,200,30);
150         opt2.setFocusable(false);
151         opt2.setBackground(Color.white);
152         this.add(opt2);
153 
154         opt3=new JRadioButton("Option 3");
155         opt3.setBounds(150,400,200,30);
156         opt3.setFocusable(false);
157         opt3.setBackground(Color.white);
158         this.add(opt3);
159 
160         opt4=new JRadioButton("Option 4");
161         opt4.setBounds(150,450,200,30);
162         opt4.setFocusable(false);
163         opt4.setBackground(Color.white);
164         this.add(opt4);
165 
166         //creating next and submit buttons
167         next=new JButton("NEXT");
168         next.setBounds(220, 500, 130, 30);
169         next.setFocusable(false);
170         next.setFont(new Font(null,Font.ITALIC,15));
171         next.setBorder(BorderFactory.createEtchedBorder());
172         next.addActionListener(this);
173         this.add(next);
174 
175         submit=new JButton("SUBMIT");
176         submit.setBounds(420, 500, 130, 30);
177         submit.setFocusable(false);
178         submit.setFont(new Font(null,Font.ITALIC,15));
179         submit.setBorder(BorderFactory.createEtchedBorder());
180         submit.addActionListener(this);
181         this.add(submit);
182         submit.setEnabled(false);//making submit button disabled only enabled in last question
183 
```

```
184         //adding radiobuttons to buttongroup
185         group=new ButtonGroup();
186         group.add(opt1);
187         group.add(opt2);
188         group.add(opt3);
189         group.add(opt4);
190 
191         //Starting the method which starts displaying questions hence starting the quiz
192         start(count);
193 
194         //Frame Properties defining the frame
195         this.add(rules);
196         this.add(question);
197         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
198         this.setTitle("Manipal Academy of Higher Education-JAVA MISAC");
199         this.setLayout(null);
200         this.setBounds(200, 50, 1000, 650);
201         this.getContentPane().setBackground(Color.white);
202         this.setIconImage(icons.getImage());
203         this.add(Header);
204         this.add(Userinfo);
205         this.add(main);
206         this.setResizable(false);
207         this.setVisible(true);
208     }
209     //Defining the start method
210     public void start(int count)
211     {
212         //Questions and options set according to the static count variable
213         question.setText(questions[count][0]);
214         opt1.setText(questions[count][1]);
215         opt1.setActionCommand(questions[count][1]);//Setting action command for each option
216         opt2.setText(questions[count][2]);
217         opt2.setActionCommand(questions[count][2]);
218         opt3.setText(questions[count][3]);
219         opt3.setActionCommand(questions[count][3]);
220         opt4.setText(questions[count][4]);
221         opt4.setActionCommand(questions[count][4]);
222         group.clearSelection();//Clearing selection when displaying next question
```

```
223     }
224     //Implementing the actionPerformed method
225     public void actionPerformed(ActionEvent e) {
226         //if user clicks on next button
227         if(e.getSource()==next)
228         {
229             //checking if user has selected an option
230             if(group.getSelection()==null) {
231                 //no option selected so displaying message
232                 JOptionPane.showMessageDialog(null,"Please select a option first before clicking next or submit");
233             }
234             else
235             {
236                 //if answer selected giving confirmatory message
237                 int ans=JOptionPane.showConfirmDialog(null,"Once pressed next, you cannot go back?\nDo You Want to Continue","Submit Answer",JOptionPane.YES_NO_OPTION);
238                 //Stroing the user answer in the useranswers array
239                 useranswers[count][0] = group.getSelection().getActionCommand();
240 
241                 //if count is 3 then making submit button enabled as it indicates the last question
242                 if (count == 3) {
243                     next.setEnabled(false);
244                     submit.setEnabled(true);
245                     }
246 
247                 //checking if user has selected YES option for confirmatory message then procedding to next question
248                 if(ans==JOptionPane.YES_OPTION && group.getSelection()!=null) {
249                     count++;
250                     start(count);
251                 }
252             }
253 
254 
255         }
256         //if user clicks on Submit button
257         if(e.getSource()==submit)
258         {
259             //no option selected so displaying message
260             if(group.getSelection()==null) {
261                 JOptionPane.showMessageDialog(null,"Please select a option first before clicking next or submit");
262             }
263             else
264             {
265                 //if answer selected giving confirmatory message
266                 int subans=JOptionPane.showConfirmDialog(null,"Do You Want to Submit the quiz","Submit Answer",JOptionPane.YES_NO_OPTION);
267                 useranswers[count][0] = group.getSelection().getActionCommand();
268                 //after storing the answer checking the useranswers with right answers and updating score accordingly
269                 for (int i = 0; i < useranswers.length; i++) {
270                     if (useranswers[i][0].equals(answers[i][1])) {
271                         score += 10;
272                     } else {
273                         score += 0;
274                 }
275                 }
276                 //checking if user has selected YES option for confirmatory message then displaying THANK YOU
277                 if(subans==JOptionPane.YES_OPTION && group.getSelection()!=null) {
278                     JOptionPane.showMessageDialog(null,"Thank you for taking the quiz,All the Best for Results");
279                     try {
280                         Thread.sleep(1000);
281                     } catch (InterruptedException e1) {
282                         e1.printStackTrace();
283                     }
284                     //close the quiz window and opening the score window
285                     this.dispose();
286                     new Score(username, score);
287                 }
288             }
289 
290         }
291 
292     }
293 
```

3. Score Class:

https://drive.google.com/file/d/16LdY-9umQGxPldsZPbCsovPxko6a4_PE/view?usp=sharing

```
1  package javaprojectfiles;
2  //Importing all necessary packages
3  import java.awt.Color;
4  import java.awt.Font;
5  import java.awt.event.ActionEvent;
6  import java.awt.event.ActionListener;
7
8  import javax.swing.ImageIcon;
9  import javax.swing.JButton;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12
13 //Creating a Score class by extending JFrame
14 public class Score extends JFrame implements ActionListener {
15     //Declaring all necessary Components required
16     JLabel label,Userinfo;JButton exit;
17
18     //Creating a constructor and getting the username and score from the quiz windo class
19     Score(String name,int score) {
20         //getting images and setting logo
21         ImageIcon icons=new ImageIcon("download.jpg");
22         ImageIcon logo=new ImageIcon("MU.png");
23
24         //Creating a Label for logo
25         label=new JLabel(logo);
26         label.setBounds(30, 0,logo.getIconWidth(),logo.getIconHeight());
27
28         //Creating a Panel to add all the detials
29         Userinfo=new JLabel("username");
30         Userinfo.setFont(new Font("Tahoma", Font.PLAIN, 16));
31         Userinfo.setBounds(60,130, 300, 280);
32
```

```
32
33          //Setting the Userinfo to show all the detials
34          Userinfo.setText(
35              "<html>"+
36                  "Student Detials" + "<br><br><br>" +
37                  "Name: "+name + "<br><br>" +
38                  "College: Manipal University-Manipal" + "<br><br>" +
39                  "Registration Number: 220968002" + "<br><br>" +
40                  "Department: DSE" + "<br><br>" +
41                  "Score: "+score+"/50"+ "<br><br>"+
42               "<html>"
43                  );
44
45          //creating exit button and adding the ActionListener
46          exit=new JButton("Exit");
47          exit.setBounds(150, 420, 120, 30);
48          exit.setFocusable(false);
49          exit.addActionListener(this);
50
51          //Adding all the components to the frame and setting the frame properties
52          this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
53          this.setTitle("Manipal Academy of Higher Education-JAVA MISAC Result");
54          this.setLayout(null);
55          this.setBounds(500, 125, 400, 550);
56          this.getContentPane().setBackground(Color.white);
57          this.setIconImage(icons.getImage());
58          this.add(label);
59          this.add(Userinfo);
60          this.add(exit);
61          this.setResizable(false);
62          this.setVisible(true);
63      }
```

```
63      }
64      //Implementing the actionPerformed method
65      public void actionPerformed(ActionEvent e) {
66          if(e.getSource()==exit)//When the user clicks on exit button then closing the frame
67          {
68              this.dispose();
69          }
70
71      }
72
73  }
74
```

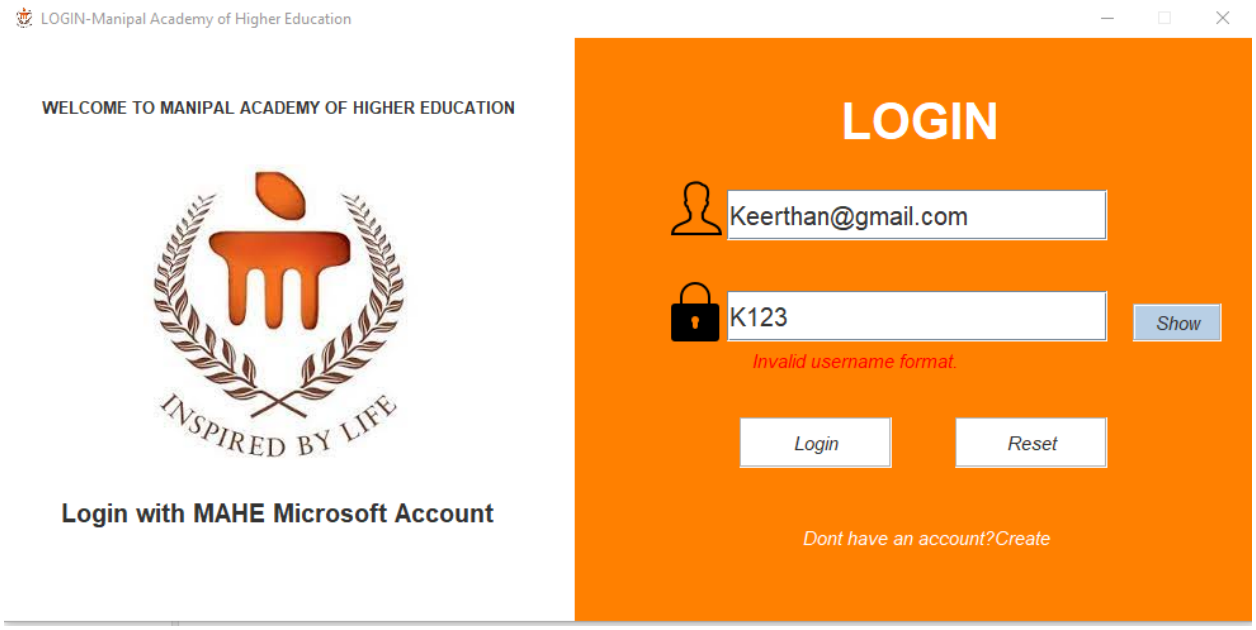
4. Main Class:

https://drive.google.com/file/d/1M3BeEXVaQfIJ3a4FLJ9CxAJPSGJjvCtc/view?usp=sharing

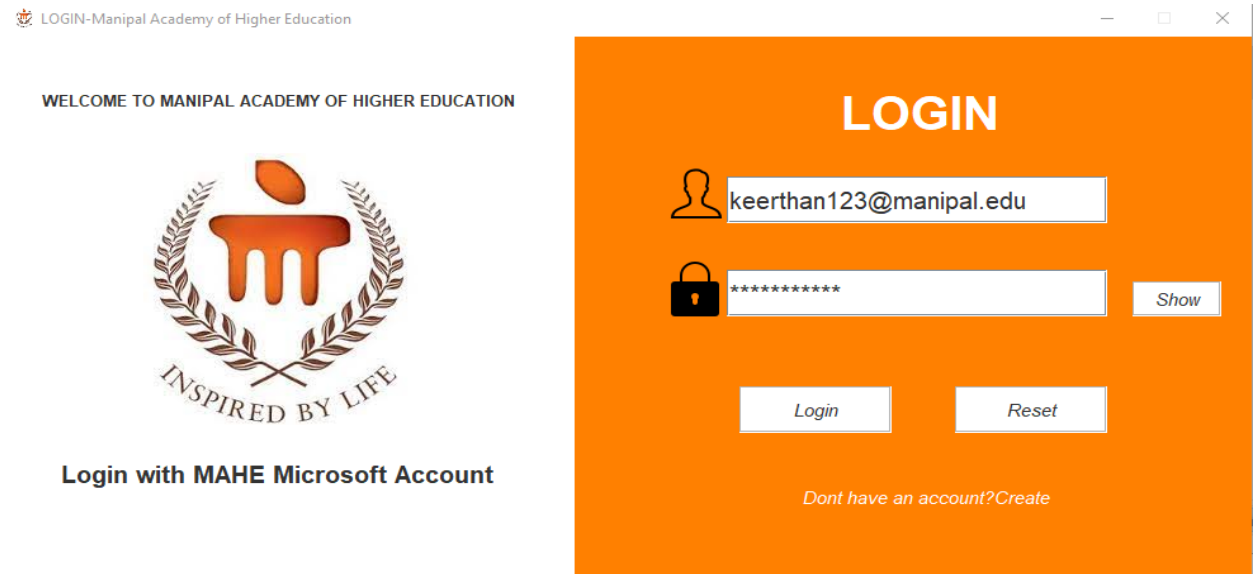
```
1  package javaprojectfiles;
2  //Creating a main class to call the LoginForm class and start the process
3  public class Main {
4
5      public static void main(String[] args) {
6          new LoginForm();
7
8      }
9
10 }
11
```

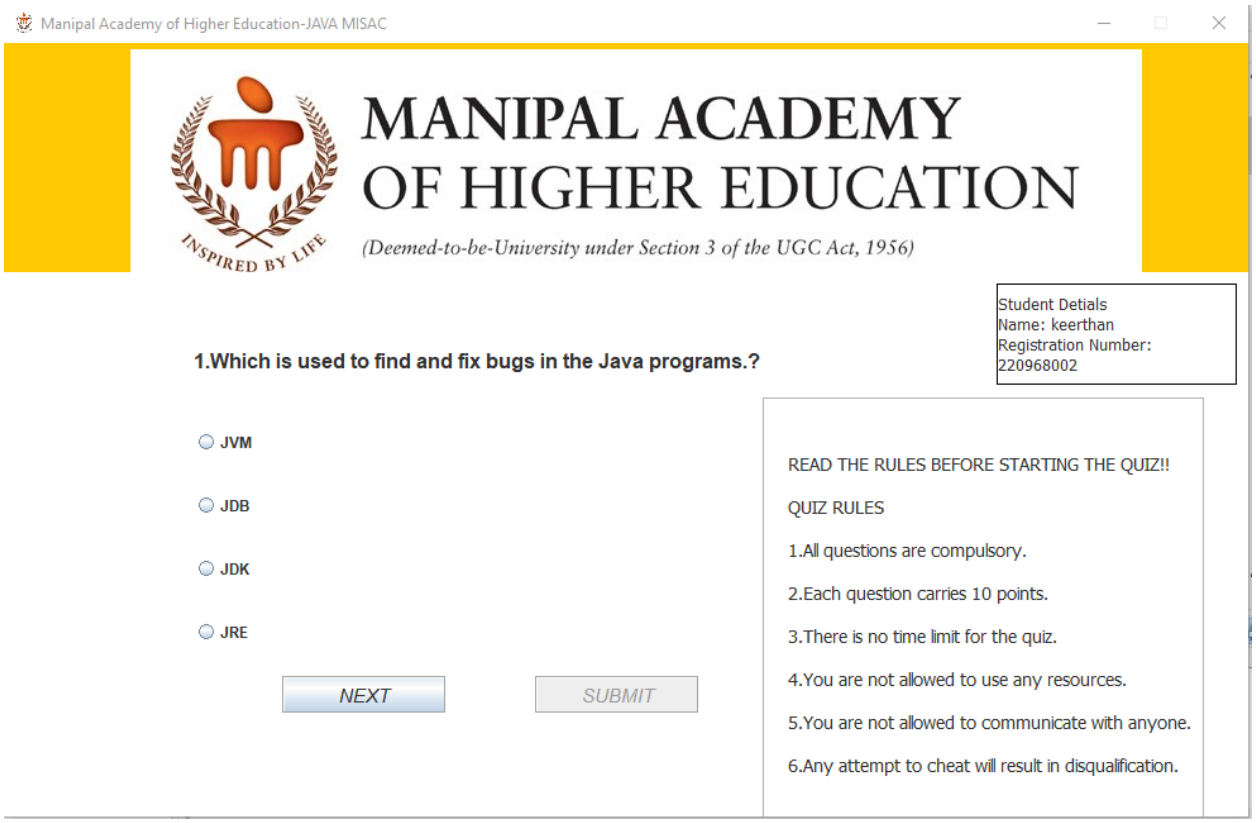
Screenshots of outputs:

1. Invalid Username and Password showing error and password shown on clicking show button.

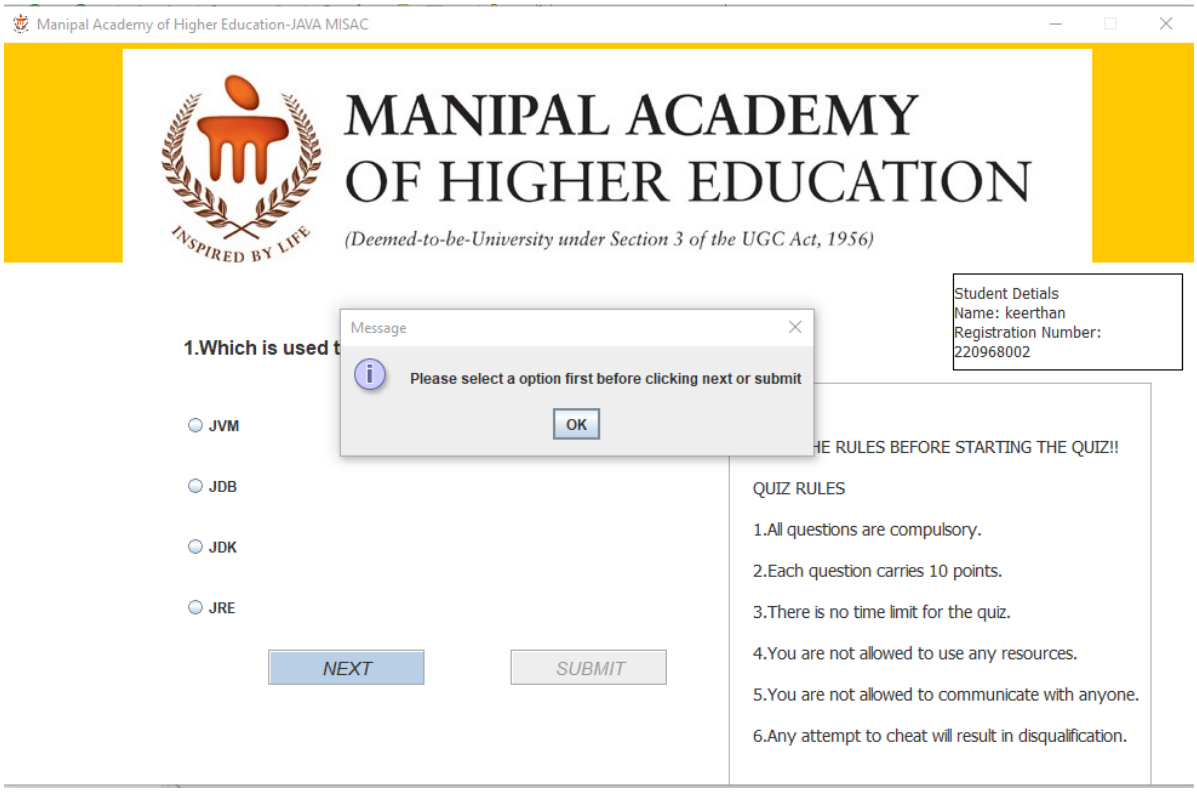


2. Proper format, on clicking the login go to the quiz window.



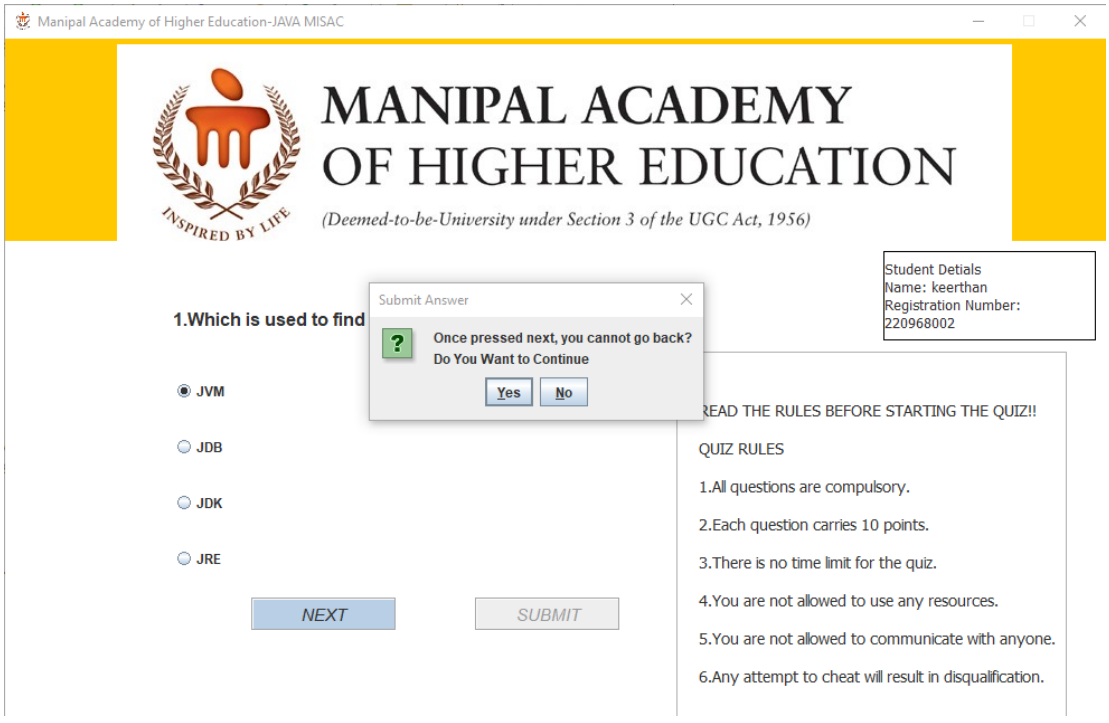


3. When no option is selected, and clicked on next shows a pop-up to select the option first.

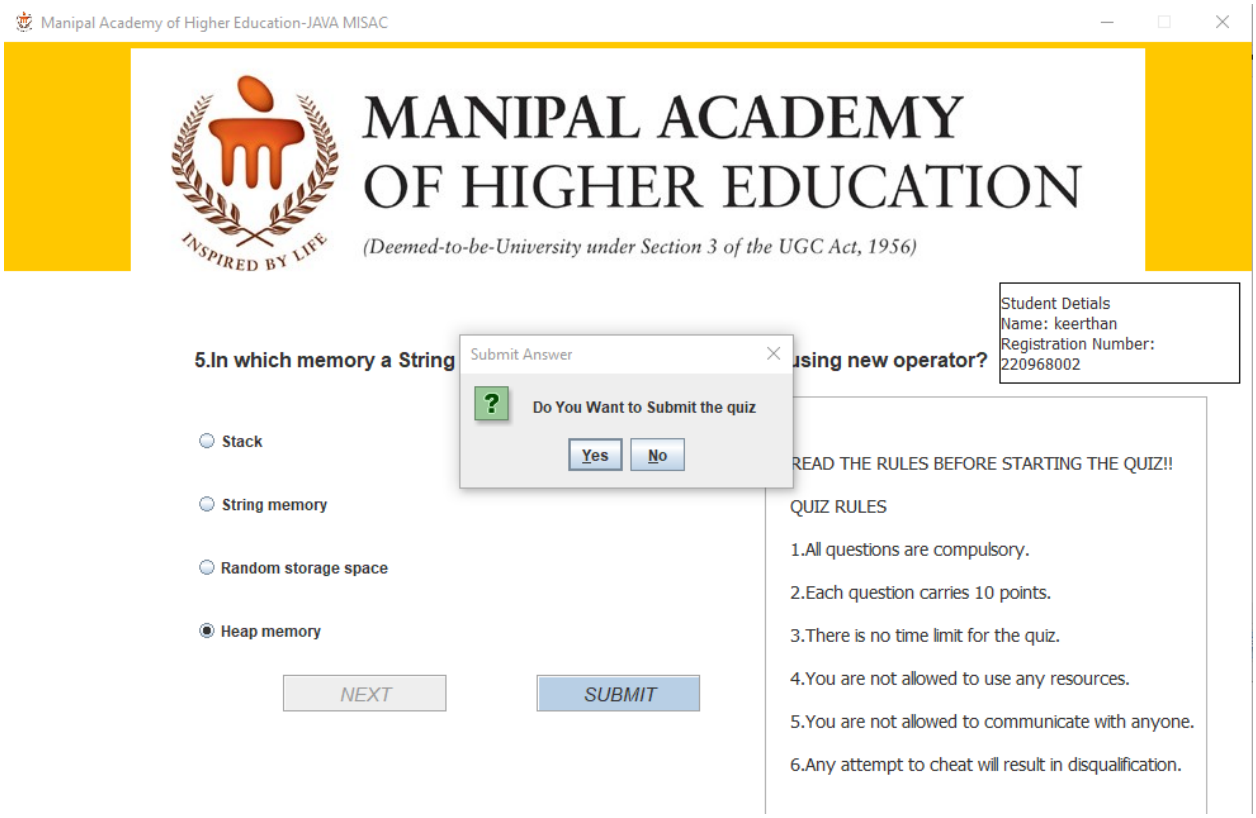


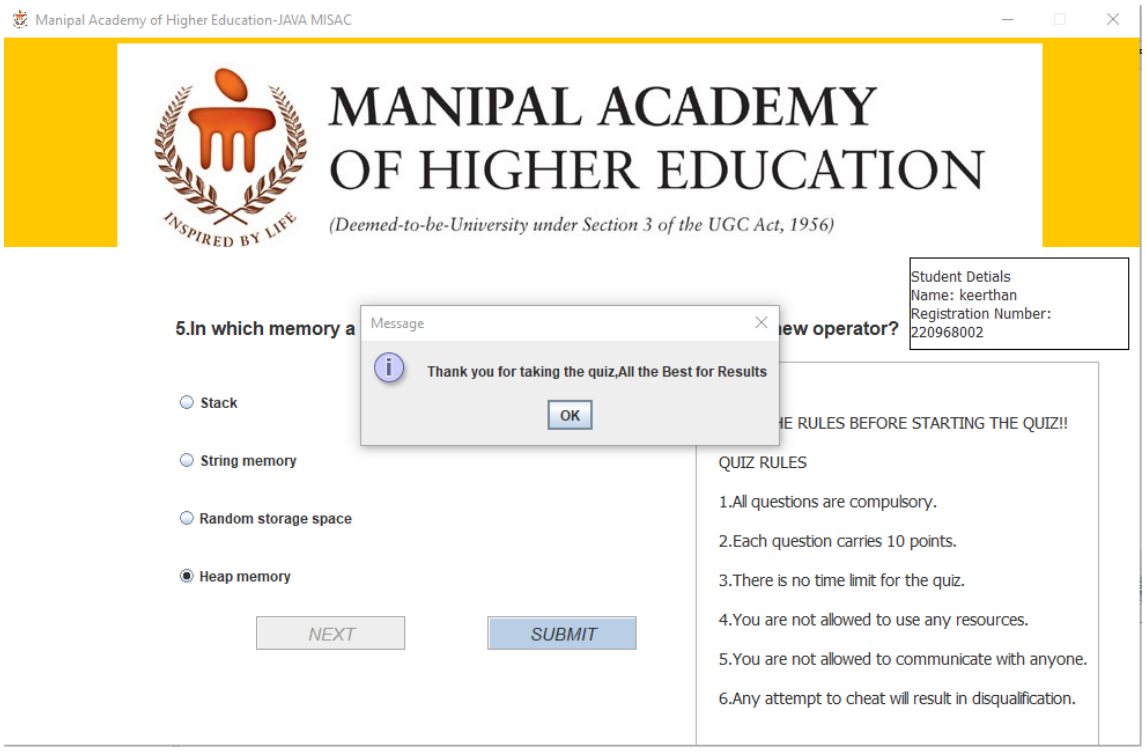
3.

4. When the option is selected and clicked on next shows a pop-up to Confirm

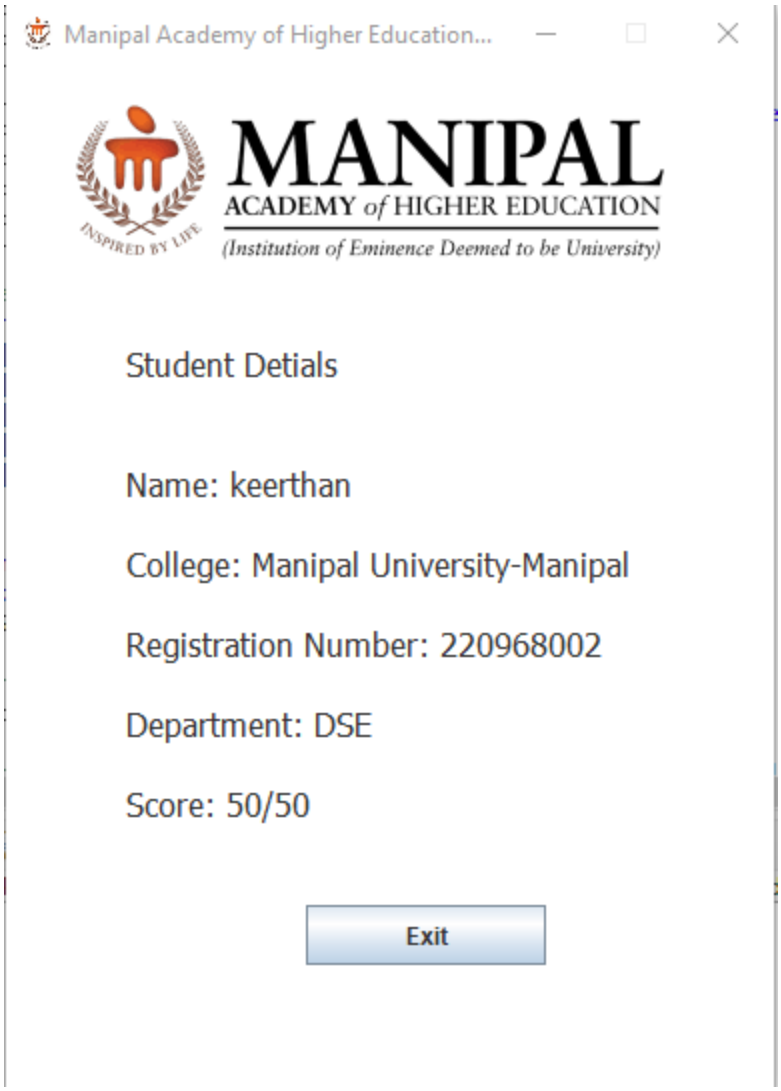


5. Once in the last question asks to confirm to submit and says Thank you





6. Once the quiz is done, the Quiz Window is closed and the Result window Opens and shows the results.



Reference:

1.https://www.coursera.org/learn/writing-java-code-for-applications

2.https://www.coursera.org/projects/build-java-gui-apps

3.https://www.youtube.com/watch?v=Kmgo00avvEw&t=11169s