

# OBJECT-ORIENTED PROGRAMMING (DSE 2123)

## MINI PROJECT SUBMISSION



Student Name: Aradhyा Goswami

Reg No: 220968034

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.

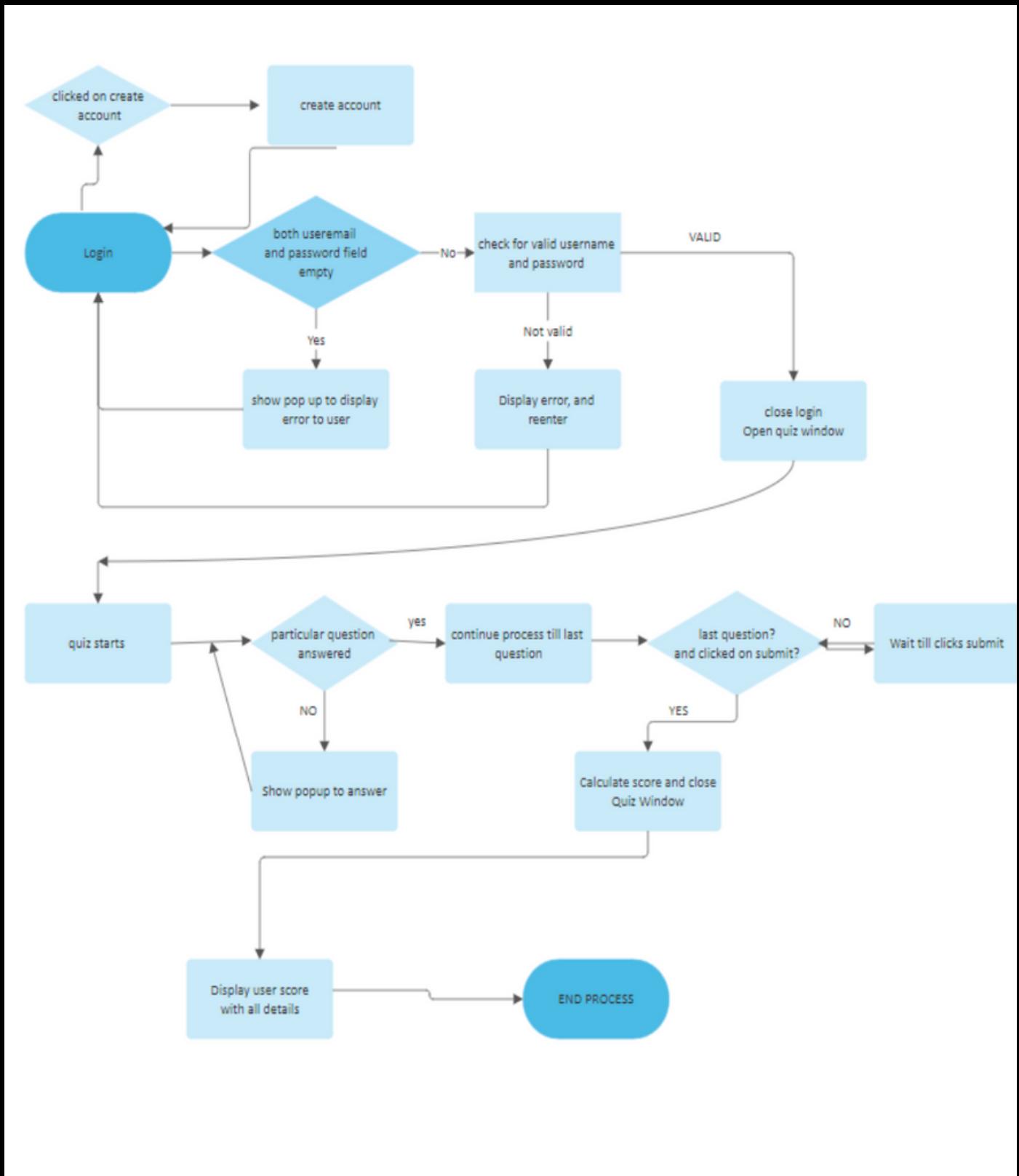
### 2. MCQ Page:

- 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 CHARTS



# COMPONENTS USED IN LOGINPANEL CLASS:

1. **JFrame**: The main frame of the application. It serves as the top-level container for other components. You can set properties such as size, title, and close operation for the frame.
2. **JTextField (username)**: Input field for the username. It allows the user to enter text, such as their username.
3. **JToggleButton (showPassword)**: Toggle button for showing/hiding the password. This button might be used to toggle between displaying the actual password characters and hiding them for security reasons.
4. **JLabel (User, Pass, Message, logo, wel, login, n, createAccountLabel)**: Labels for displaying text, icons, and messages. They are used to provide information to the user or to identify other components.
5. **JButton (Login, Reset)**: Buttons for user interaction. These buttons typically trigger actions such as submitting a form (**Login**) or resetting input fields (**Reset**).
6. **JPanel (left and right)**: Panels used for layout and design. **JPanel** is a generic container that is often used to group and organize other components. Having separate left and right panels suggests a specific layout strategy, possibly dividing the form into distinct sections for a visually appealing design.
7. **ImageIcon**: Used to load and display images. This component allows you to easily integrate images, such as logos, into your user interface.
8. **JPasswordField (password)**: Input field for the password. It is similar to **JTextField** but is specifically designed for handling password input, obscuring the entered characters.
9. **Cursor**: Used to set the cursor style for the "Create Account" label. This can be used to provide visual feedback to the user when hovering over the label, indicating that it is clickable.

# EVENTS AND ACTIONS USED IN LOGINPANEL CLASS:

1. **ItemStateChanged(ItemEvent e)**: This method is used to handle changes in the state of the Show toggle button:
  - If the user selects (toggles) the **Show** button, the password field is set to display the actual password characters.
  - If the user deselects the **Show** button, the password field is set to hide the password characters for security.
2. **actionPerformed(ActionEvent e)**: This method is responsible for handling actions triggered by button clicks in the **LoginPanel** class. Specifically:
  - If the user clicks the **Reset** button, it will clear the username and password fields, as well as any displayed error messages.
  - If the user clicks the **Login** button, the method performs the following actions:
    - Checks if the entered username and password are in the correct format.
    - If the input is not valid, an error message is displayed.
    - If the input is valid, the login window is closed, and a new quiz window is opened with the provided username.

The **LoginPanel** class employs a variety of components and event-handling mechanisms to construct an effective login window for the project. Crucial to this functionality are the 'isUsernameValid' and 'isPasswordValid' methods, which validate the format of the username and password, respectively, ensuring they satisfy specific criteria.

This **LoginPanel** class serves as the foundational element for user authentication, acting as a prerequisite before users can access the quiz section of the application. The integration of various components, event handling, and validation methods ensures the creation of a robust and secure login mechanism within the broader project framework.

## COMPONENTS USED IN QUIZWINDOW CLASS:

- **JFrame:** Represents the main frame of the application.
- **JLabel (RULEs, Userinfo, label, question):** Labels used for displaying text and images. These labels provide information, instructions, and content within the quiz panel.
- **ButtonGroup:** Groups radio buttons (opt1, opt2, opt3, opt4) to ensure exclusive selection. This ensures that users can only choose one answer option at a time.
- **JPanel (rules, Header, main):** Panels utilized for layout and design. These panels help organize and structure the visual elements within the quiz panel.
- **JButton (next, submit):** Buttons designed for user interaction. The "Next" button typically advances to the next question, while the "Submit" button allows users to finalize their answers.
- **JOptionPane:** Displays dialog boxes for user prompts and messages. This component is handy for presenting information or requesting input from the user during the quiz.
- **JRadioButton (opt1, opt2, opt3, opt4):** Radio buttons for selecting answer options. Users can choose from these options to provide their answers to quiz questions.
- **ImageIcon:** Used to load and display images. This component enhances the visual experience by incorporating images into the quiz panel.

## EVENT HANDLING IN QUIZWINDOW CLASS:

1. **actionPerformed(ActionEvent e):** This method manages actions triggered by button clicks in the **QuizPanel1** class.

- **"NEXT" Button:**
  - Checks if an option is selected.
  - If an option is selected, it prompts the user 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.
- **"SUBMIT" Button:**
  - Checks if an option is selected.
  - If an option is selected, it prompts the user for confirmation to submit the quiz.
  - If the user confirms and selects an option, it calculates the score and opens a new window displaying the score.

## COMPONENTS UTILIZED IN SCORE CLASS:

1. **JFrame**: This serves as the central framework for the application, providing the primary structure for the Score class.
2. **JLabel (label, Userinfo)**: These labels play a pivotal role in presenting textual information and images within the Score class. They contribute to the informative and visual aspects of the user interface.
3. **JButton (exit)**: The exit button serves as a user interface element, allowing users to interact with the application. It likely provides functionality for closing or exiting the Score window.
4. **ImageIcon**: The ImageIcon component is employed to load and display images. This enhances the visual appeal of the Score class, potentially incorporating graphical elements to enrich the user experience.

## EVENTS AND ACTIONS IN SCORE CLASS:

1. **actionPerformed(ActionEvent e)**: This method orchestrates responses to button clicks within the Score class.
  - "Exit" Button:
    1. When the user clicks the "Exit" button, the method takes action by disposing of the current frame, effectively concluding the application.

The Score class is designed as the score window in your Java GUI application, providing a comprehensive overview of user details and quiz performance upon completion. Notably, the "Exit" button is strategically placed to offer users the option to gracefully close the result window. Details such as the user's name, college, registration number, department, and the obtained score out of 50 are thoughtfully presented within this informative window.

# PROGRAM CODE

## 1. LoginForm Class:

DriveLinkForLginFormClass

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

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

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

```
97     Message=new JLabel();
98     Message.setBounds(140,245,350,20);
99     Message.setForeground(Color.RED);
100    Message.setFont(new Font(null,Font.ITALIC,15));
101    right.add(Message);
102    Login=new JButton("Login");
103    Login.setBounds(130,300,120,40);
104    Login.setFocusable(false);
105    Login.addActionListener(this);
106    Login.setBackground(Color.white);
107    Login.setFont(new Font(null,Font.ITALIC,15));
108    Login.setBorder(BorderFactory.createEtchedBorder());
109    right.add(Login);
110    Reset=new JButton("Reset");
111    Reset.setBounds(300,300,120,40);
112    Reset.setFocusable(false);
113    Reset.addActionListener(this);
114    Reset.setBackground(Color.white);
115    Reset.setFont(new Font(null,Font.ITALIC,15));
116    Reset.setBorder(BorderFactory.createEtchedBorder());
117    right.add(Reset);
118    showPassword = new JToggleButton("Show");
119    showPassword.setBounds(440, 210, 70, 30);
120    showPassword.setFocusable(false);
121    showPassword.addItemListener(this);
122    showPassword.setBackground(Color.white);
123    showPassword.setFont(new Font(null,Font.ITALIC,15));
124    showPassword.setBorder(BorderFactory.createEtchedBorder());
125    right.add(showPassword);
126
```

```
127    createAccountLabel.setBounds(180, 380, 350, 30);
128    createAccountLabel.setForeground(Color.WHITE);
129    createAccountLabel.setFont(new Font("Peterdraw",Font.ITALIC,15));
130    createAccountLabel.setCursor(Cursor.getPredefinedCursor(Cursor.HAND_CURSOR));
131    createAccountLabel.addMouseListener(new MouseAdapter() {
132        public void mouseClicked(MouseEvent e) {
133            try {
134                Desktop.getDesktop().browse(new URI("https://signup.live.com/"));
135            } catch (IOException ex) {
136                ex.printStackTrace();
137            } catch (URISyntaxException e1) {
138                e1.printStackTrace();
139            }
140        }
141    });
142    right.add(createAccountLabel);
143
144    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
145    this.setLayout(null);
146    this.setBounds(200, 100, 1000, 500);
147    this.setTitle("LOGIN-Manipal Academy of Higher Education");
148    this.setIconImage(icon.getImage());
149    this.setBackground(Color.white);
150    this.setResizable(false);
151    this.setVisible(true);
152}
153
154}
```

```

155 ⊕ public void actionPerformed(ActionEvent e) {
156     if(e.getSource()==Reset)
157     {
158         username.setText("");
159         password.setText("");
160         Message.setText("");
161     }
162     if(e.getSource()==Login)
163     {
164         String log=username.getText();
165         String pas=String.valueOf(password);
166         if(log.isEmpty() || pas.isEmpty())
167         {
168             Message.setText("Please enter a Valid Input");
169         }
170         else
171         {
172             if (!isValidUsername(log) && !isValidPassword(pas)) {
173                 Message.setText("Invalid username and password format.");
174             } else if (!isValidUsername(log)) {
175                 Message.setText("Invalid username format.");
176             } else if(!isValidPassword(pas)){
177                 Message.setText("Invalid password format.");
178             }
179             else
180             {
181                 this.dispose();
182                 new quizWindow(log);
183             }
184         }
185     }
186 }
187 }
```

```

188 ⊕ private boolean isValidUsername(String username) {
189     // Username should contain "@" and end with ".com" and be all in lowercase.
190     String regex = "^[a-z0-9]+@[.]*\\.[a-z]{2,}$";
191     return Pattern.matches(regex, username);
192 }
193
194 // Helper method to check if the password is in the correct format
195 ⊕ private boolean isValidPassword(String password) {
196     // Password should contain at least one uppercase letter and one digit
197     return Pattern.compile("^(?=.*[A-Z])(?=.*[\\d]).+$").matcher(password).matches();
198 }
199
200 ⊕ public void itemStateChanged(ItemEvent e) {
201     if (e.getStateChange() == ItemEvent.SELECTED) {
202         password.setEchoChar((char) 0); // Show the password as a string (no masking)
203     } else {
204         password.setEchoChar('*'); // Mask the password
205     }
206 }
207
208 }
209 }
```

## 2. QuizWindow Class:

DriveLinkForQuizWindowClass

```
1 package JAVA_MiniProject;
2
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import javax.swing.*;
7 import javax.swing.border.Border;
8 import javax.swing.border.EtchedBorder;
9
10 public class QuizPanel1 extends JFrame implements ActionListener {
11     ImageIcon icons, logo;
12     JRadioButton opt1, opt2, opt3, opt4;
13     ButtonGroup group;
14     JLabel label, question, RULEs, Userinfo;
15     JButton next, submit;
16     JPanel rules, Header, main;
17
18     String questions[][] = new String[5][5];
19     {
20         questions[0][0] = "1. Which is used to find and fix bugs in the Java Programs?";
21         questions[0][1] = "JVM";
22         questions[0][2] = "JDB";
23         questions[0][3] = "JDK";
24         questions[0][4] = "JRE";
25
26         questions[1][0] = "2. What is the return type of the hashCode() method in the Object Class?";
27         questions[1][1] = "int";
28         questions[1][2] = "Object";
29         questions[1][3] = "long";
30         questions[1][4] = "void";
31
32         questions[2][0] = "3. Which package contains the Random Class?";
33         questions[2][1] = "java.util package";
34         questions[2][2] = "java.lang package";
35         questions[2][3] = "java.awt package";
36         questions[2][4] = "java.io package";
37
38         questions[3][0] = "4. An interface with no fields or methods is known as ?";
39         questions[3][1] = "Runnable Interface";
40         questions[3][2] = "Abstract Interface";
41         questions[3][3] = "Marker Interface";
42         questions[3][4] = "CharSequence Interface";
```

```

43     questions[4][0] = "5. In which memory a String is stored when we create a string using the new operator?";
44     questions[4][1] = "Stack";
45     questions[4][2] = "String Memory";
46     questions[4][3] = "Random Storage Space";
47     questions[4][4] = "Heap Memory";
48   }
49   String answers[][] = new String[5][2];
50   {
51     answers[0][1] = "JDB";
52     answers[1][1] = "int";
53     answers[2][1] = "java.util package";
54     answers[3][1] = "Marker Interface";
55     answers[4][1] = "Heap memory";
56   }
57
58   String useranswers[][] = new String[5][1];
59   static int count = 0;
60   int score = 0;
61   String username;
62
63
68 quizWindow(String name)
69 {
70   username=name;
71   int ind=username.indexOf('@');
72   username=username.substring(0,ind);
73   ImageIcon icons=new ImageIcon("download.jpg");
74   ImageIcon logo=new ImageIcon("m.jpg");
75   label=new JLabel(logo);
76   label.setBounds(90, 0,logo.getIconWidth(),logo.getIconHeight());
77
78   Header=new JPanel();
79   Header.setBounds(0,0,1000,logo.getIconHeight());
80   Header.setBackground(Color.orange);
81   Header.add(label);
82
83   question=new JLabel();
84   question.setBounds(150,200,700,100);
85   question.setFont(new Font("Peterdraw",Font.BOLD,16));
86
87   main=new JPanel();
88   main.setBounds(100,logo.getIconHeight()+10,logo.getIconWidth() ,400);
89   main.setBackground(Color.white);
90
91   rules=new JPanel();
92   rules.setBounds(600, 280, 350, 350);
93   rules.setBackground(new Color(255,255,255));
94   rules.setBorder(BorderFactory.createEtchedBorder());
95   rules.setLayout(null);
96   rules.setFont(new Font("Tahoma", Font.PLAIN, 16));
97   RULEs=new JLabel();
98   RULEs.setBounds(20, 5, 700, 350);
99   RULEs.setFont(new Font("Tahoma", Font.PLAIN, 14));
100  RULEs.setText(
101    "<html>" +
102      "READ THE RULES BEFORE STARTING THE QUIZ!!" + "<br><br>" +
103      "QUIZ RULES" + "<br><br>" +
104      "1.All questions are compulsory." + "<br><br>" +
105      "2.Each question carries 10 points." + "<br><br>" +
106      "3.There is no time limit for the quiz." + "<br><br>" +
107      "4.You are not allowed to use any resources." + "<br><br>" +
108      "5.You are not allowed to communicate with anyone." + "<br><br>" +
109      "6.Any attempt to cheat will result in disqualification." + "<br><br>" +
110

```

```

111         "<html>"  

112     );  

113     rules.add(RULEs);  

114  

115     Userinfo=new JLabel(username);  

116     Userinfo.setFont(new Font("Tahoma", Font.PLAIN, 13));  

117     Userinfo.setBounds(785, 190, 190, 80);  

118     Userinfo.setBorder(BorderFactory.createLineBorder(Color.black));  

119     Userinfo.setText(  

120         "<html>" +  

121             "Student Detials" + "<br>" +  

122             "Name: "+username + "<br>" +  

123             "Registration Number: 220968034" + "<br>" +  

124             "<html>"  

125         );  

126     opt1=new JRadioButton("Option 1");  

127     opt1.setBounds(150,300,200,30);  

128     opt1.setFocusable(false);  

129     opt1.setBackground(Color.white);  

130     this.add(opt1);  

131  

132     opt2=new JRadioButton("Option 2");  

133     opt2.setBounds(150,350,200,30);  

134     opt2.setFocusable(false);  

135     opt2.setBackground(Color.white);  

136     this.add(opt2);  

137  

138     opt3=new JRadioButton("Option 3");  

139     opt3.setBounds(150,400,200,30);  

140     opt3.setFocusable(false);  

141     opt3.setBackground(Color.white);  

142     this.add(opt3);  

143  

144     opt4=new JRadioButton("Option 4");  

145     opt4.setBounds(150,450,200,30);  

146     opt4.setFocusable(false);  

147     opt4.setBackground(Color.white);  

148     this.add(opt4);  

149

```

```

150     next=new JButton("NEXT");  

151     next.setBounds(220, 500, 130, 30);  

152     next.setFocusable(false);  

153     next.setFont(new Font(null,Font.ITALIC,15));  

154     next.setBorder(BorderFactory.createEtchedBorder());  

155     next.addActionListener(this);  

156     this.add(next);  

157  

158     submit=new JButton("SUBMIT");  

159     submit.setBounds(420, 500, 130, 30);  

160     submit.setFocusable(false);  

161     submit.setFont(new Font(null,Font.ITALIC,15));  

162     submit.setBorder(BorderFactory.createEtchedBorder());  

163     submit.addActionListener(this);  

164     this.add(submit);  

165     submit.setEnabled(false);  

166     group=new ButtonGroup();  

167     group.add(opt1);  

168     group.add(opt2);  

169     group.add(opt3);  

170     group.add(opt4);  

171  

172     start(count);  

173     this.add(rules);  

174     this.add(question);  

175     this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  

176     this.setTitle("Manipal Academy of Higher Education-JAVA MISAC");  

177     this.setLayout(null);  

178     this.setBounds(200, 50, 1000, 650);  

179     this.getContentPane().setBackground(Color.white);  

180     this.setIconImage(Icons.getImage());  

181     this.add(header);  

182     this.add(Userinfo);  

183     this.add(main);  

184     this.setResizable(false);  

185     this.setVisible(true);  

186 }

```

```

187     public void start(int count)
188     {
189         question.setText(questions[count][0]);
190         opt1.setText(questions[count][1]);
191         opt1.setActionCommand(questions[count][1]);
192         opt2.setText(questions[count][2]);
193         opt2.setActionCommand(questions[count][2]);
194         opt3.setText(questions[count][3]);
195         opt3.setActionCommand(questions[count][3]);
196         opt4.setText(questions[count][4]);
197         opt4.setActionCommand(questions[count][4]);
198         group.clearSelection();
199     }
200     public void actionPerformed(ActionEvent e) {
201         if(e.getSource()==next)
202         {
203             if(group.getSelection()==null) {
204                 JOptionPane.showMessageDialog(null,"Please select a option first before clicking next or submit");
205             }
206             else
207             {
208                 int ans=JOptionPane.showConfirmDialog(null,"Once pressed next, you cannot go back?\nDo You "
209                     + "Want to Continue","Submit Answer",JOptionPane.YES_NO_OPTION);
210                 useranswers[count][0] = group.getSelection().getActionCommand();
211
212                 if (count == 3) {
213                     next.setEnabled(false);
214                     submit.setEnabled(true);
215                 }
216
217
218                 if(ans==JOptionPane.YES_OPTION && group.getSelection()!=null) {
219                     count++;
220                     start(count);
221                 }
222             }
223         }
224     }
225
226     if(e.getSource()==submit)
227     {
228         if(group.getSelection()==null) {
229             JOptionPane.showMessageDialog(null,"Please select a option first before clicking next or submit");
230         }
231         else
232         {
233             int subans=JOptionPane.showConfirmDialog(null,"Do You Want to Submit the quiz","Submit Answer"
234                     ,JOptionPane.YES_NO_OPTION);
235             useranswers[count][0] = group.getSelection().getActionCommand();
236             for (int i = 0; i < useranswers.length; i++) {
237                 if (useranswers[i][0].equals(answers[i][1])) {
238                     score += 10;
239                 } else {
240                     score += 0;
241                 }
242                 if(subans==JOptionPane.YES_OPTION && group.getSelection()!=null) {
243                     JOptionPane.showMessageDialog(null,"Thank you for taking the quiz,All the Best for Results");
244                     try {
245                         Thread.sleep(1000);
246                     } catch (InterruptedException e1) {
247                         e1.printStackTrace();
248                     }
249                     this.dispose();
250                     new Score(username, score);
251                 }
252             }
253         }
254     }
255
256 }
257 }
```

### 3. Score Class:

[DriveLinkForScorePage](#)

```
1 package JAVA_MiniProject;
2
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 public class Score extends JFrame implements ActionListener {
14     JLabel label,Userinfo;JButton exit;
15
16     Score(String name,int score) {
17         ImageIcon icons=new ImageIcon("download.jpg");
18         ImageIcon logo=new ImageIcon("MU.png");
19
20         label=new JLabel(logo);
21         label.setBounds(30, 0,logo.getIconWidth(),logo.getIconHeight());
22
23         Userinfo=new JLabel("username");
24         Userinfo.setFont(new Font("Tahoma", Font.PLAIN, 16));
25         Userinfo.setBounds(60,130, 300, 280);
26
27         Userinfo.setText(
28             "<html>" +
29                 "Student Detials" + "<br><br><br>" +
30                 "Name: "+name + "<br><br>" +
31                 "College: Manipal University-Manipal" + "<br><br>" +
32                 "Registration Number: 220968034" + "<br><br>" +
33                 "Department: DSE" + "<br><br>" +
34                 "Score: "+score+"/"+50+"<br><br>" +
35                 "</html>" +
36             );
37
38         exit=new JButton("Exit");
39         exit.setBounds(150, 420, 120, 30);
40         exit.setFocusable(false);
41         exit.addActionListener(this);
42
43         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
44         this.setTitle("Manipal Academy of Higher Education-JAVA MISAC Result");
45         this.setLayout(null);
46         this.setBounds(500, 125, 400, 550);
47         this.getContentPane().setBackground(Color.white);
48         this.setIconImage(icons.getImage());
49         this.add(label);
50         this.add(Userinfo);
51         this.add(exit);
52         this.setResizable(false);
53         this.setVisible(true);
54     }
55     public void actionPerformed(ActionEvent e) {
56         if(e.getSource()==exit)
57         {
58             this.dispose();
59         }
60     }
61 }
62 public static void main(String[] args) {
63     new Score("",0);
64 }
65 }
66 }
```

### 3. Main Class:

DriveLinkForMainClass

```
1 package JAVA_MiniProject;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         new LoginPanel();
8     }
9 }
10 |
```

## OUTPUTS:

- Upon entering an Invalid username or password, an error message is displayed to notify the user of the INVALID input. Additionally, there is a "Show" button functionality, allowing the user to reveal the entered password characters upon clicking.

The screenshot shows the Manipal Academy of Higher Education login page. The top header reads "LOGIN-Manipal Academy of Higher Education". Below it, the welcome message "WELCOME TO MANIPAL ACADEMY OF HIGHER EDUCATION" is displayed, along with the MAHE logo and the tagline "INSPIRED BY LIFE". A "Login with MAHE Microsoft Account" button is present. On the right, a teal-colored "LOGIN" form is shown. In the "Username" field, "Aradhya814" is entered. Below the "Username" field, an error message "Invalid username format." is displayed. To the right of the "Username" field is a "Show" button. At the bottom of the form are "Login" and "Reset" buttons, and a link "Dont have an account?Create".

- Upon entering an Invalid username or password, an error message is displayed to notify the user of the INVALID input. Additionally, there is a "Show" button functionality, allowing the user to reveal the entered password characters upon clicking.

The screenshot shows the Manipal Academy of Higher Education login page. The top header reads "LOGIN-Manipal Academy of Higher Education". Below it, the welcome message "WELCOME TO MANIPAL ACADEMY OF HIGHER EDUCATION" is displayed, along with the MAHE logo and the tagline "INSPIRED BY LIFE". A "Login with MAHE Microsoft Account" button is present. On the right, a teal-colored "LOGIN" form is shown. In the "Username" field, "aradhya814@gmail.com" is entered. Below the "Username" field is a password input field containing masked password characters ("\*\*\*\*\*"). To the right of the password field is a "Show" button. At the bottom of the form are "Login" and "Reset" buttons, and a link "Dont have an account?Create".

Manipal Academy of Higher Education-JAVA MISAC



# MANIPAL ACADEMY OF HIGHER EDUCATION

(Deemed-to-be-University under Section 3 of the UGC Act, 1956)

Student Details  
Name: Aradhyा  
Registration Number:  
220968034

1.Which is used to find and fix bugs in the Java programs.?

JVM  
 JDB  
 JDK  
 JRE

**NEXT**      **SUBMIT**

READ THE RULES BEFORE STARTING THE QUIZ!!

QUIZ RULES

- 1.All questions are compulsory.
- 2.Each question carries 10 points.
- 3.There is no time limit for the quiz.
- 4.You are not allowed to use any resources.
- 5.You are not allowed to communicate with anyone.
- 6.Any attempt to cheat will result in disqualification.

3. If no option is chosen and the "Next" button is clicked, a pop-up prompts the user to select an option before proceeding.

Manipal Academy of Higher Education-JAVA MISAC



# MANIPAL ACADEMY OF HIGHER EDUCATION

(Deemed-to-be-University under Section 3 of the UGC Act, 1956)

Student Details  
Name: Aradhyा  
Registration Number:  
220968034

1.Which is used to find and fix bugs in the Java programs.?

JVM  
 JDB  
 JDK  
 JRE

**NEXT**      **SUBMIT**

Message

Please select a option first before clicking next or submit

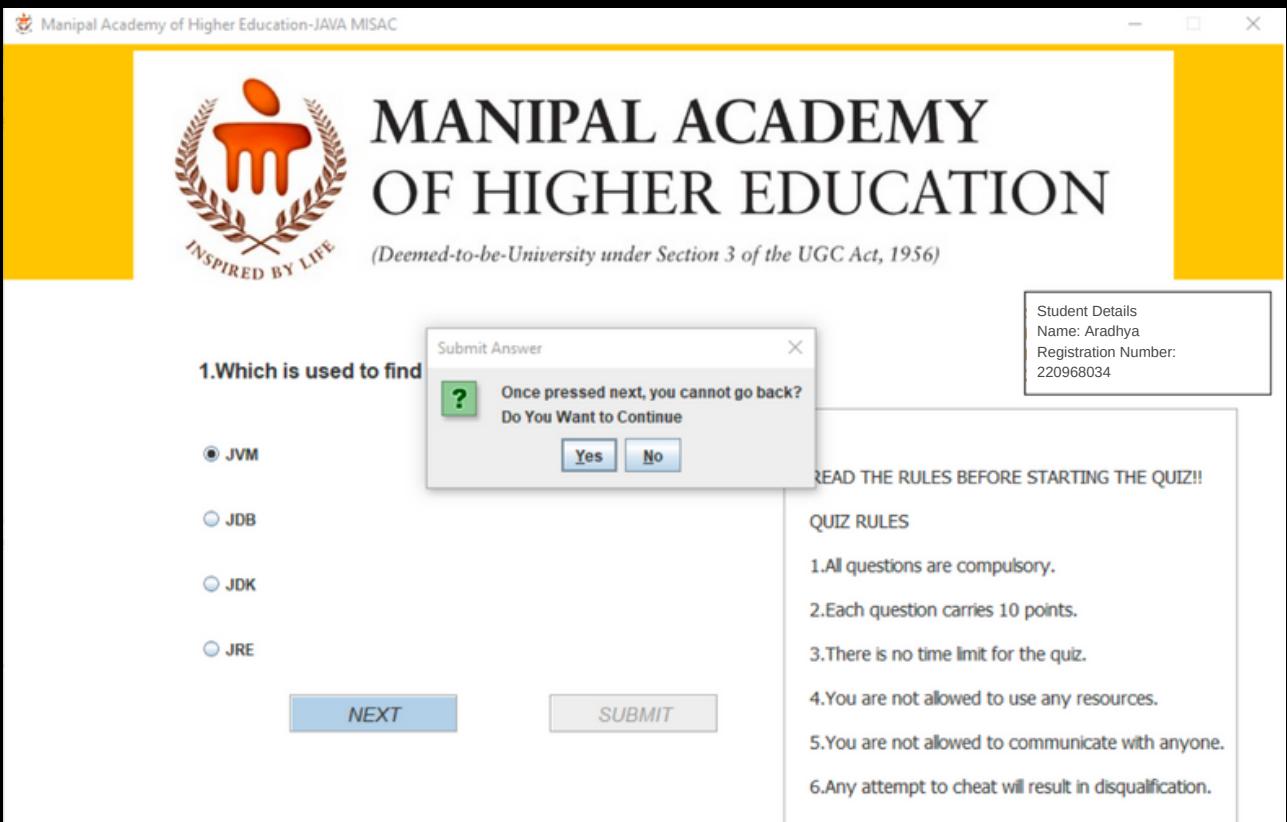
**OK**

READ THE RULES BEFORE STARTING THE QUIZ!!

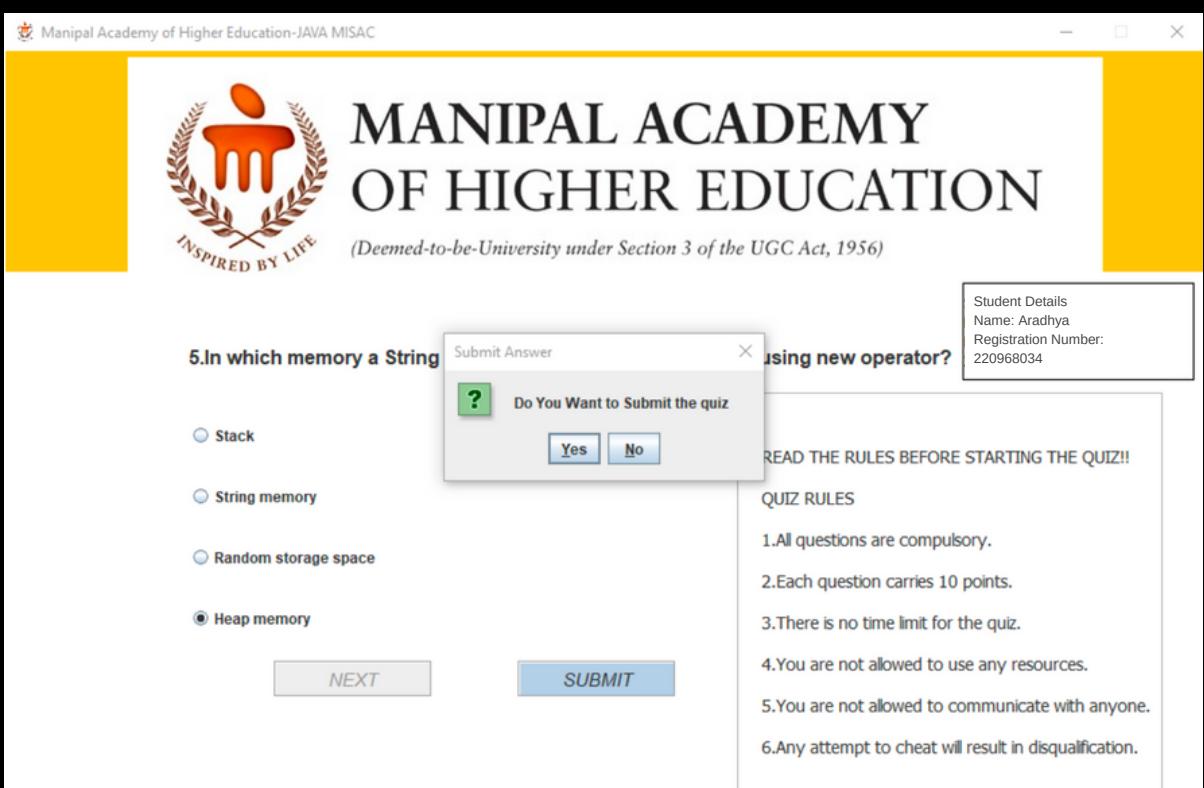
QUIZ RULES

- 1.All questions are compulsory.
- 2.Each question carries 10 points.
- 3.There is no time limit for the quiz.
- 4.You are not allowed to use any resources.
- 5.You are not allowed to communicate with anyone.
- 6.Any attempt to cheat will result in disqualification.

4. After choosing an option, clicking the NEXT button prompts a confirmation pop-up.



5. Upon reaching the last question, a confirmation is requested to submit, and upon confirmation, a message expresses gratitude by saying "**THANK YOU.**"



Manipal Academy of Higher Education-JAVA MISAC



# MANIPAL ACADEMY OF HIGHER EDUCATION

(Deemed-to-be-University under Section 3 of the UGC Act, 1956)

Student Details  
Name: Aradhya  
Registration Number:  
220968034

5.In which memory a new operator?

Stack  
 String memory  
 Random storage space  
 Heap memory

Message  
Thank you for taking the quiz, All the Best for Results  
OK

RE RULES BEFORE STARTING THE QUIZ!!

QUIZ RULES

- 1.All questions are compulsory.
- 2.Each question carries 10 points.
- 3.There is no time limit for the quiz.
- 4.You are not allowed to use any resources.
- 5.You are not allowed to communicate with anyone.
- 6.Any attempt to cheat will result in disqualification.

NEXT SUBMIT

6. Upon completing the quiz, the QUIZ WINDOW is closed, and the RESULT WINDOW opens, displaying the results.

Manipal Academy of Higher Education...



# MANIPAL ACADEMY of HIGHER EDUCATION

(Institution of Eminence Deemed to be University)

Student Details

Name: Aradhya

College: Manipal University-Manipal

Registration Number: 220968034

Department: DSE

Score: 50/50

Exit

# REFERENCES

- 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>