## horizontal line



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.

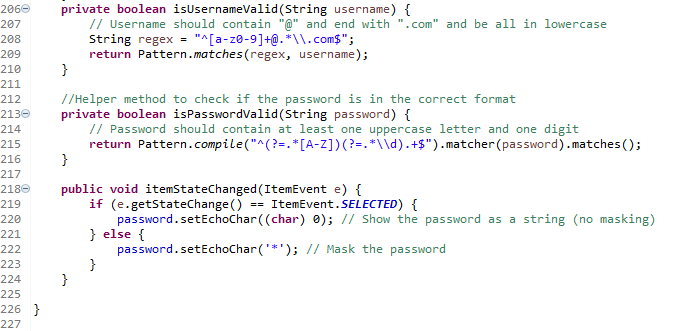
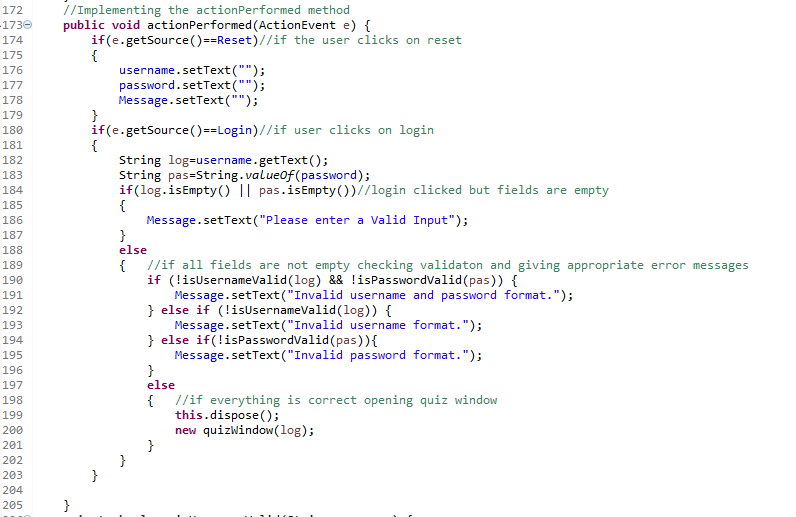
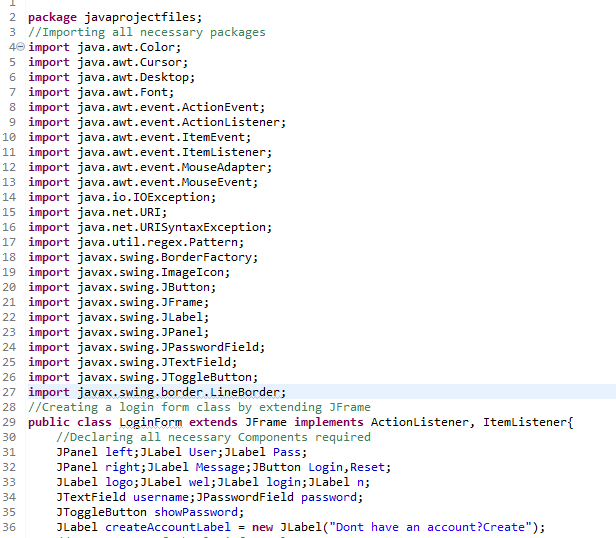
- If the "Exit" button is clicked, it disposes of the current frame, effectively closing the application.

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:

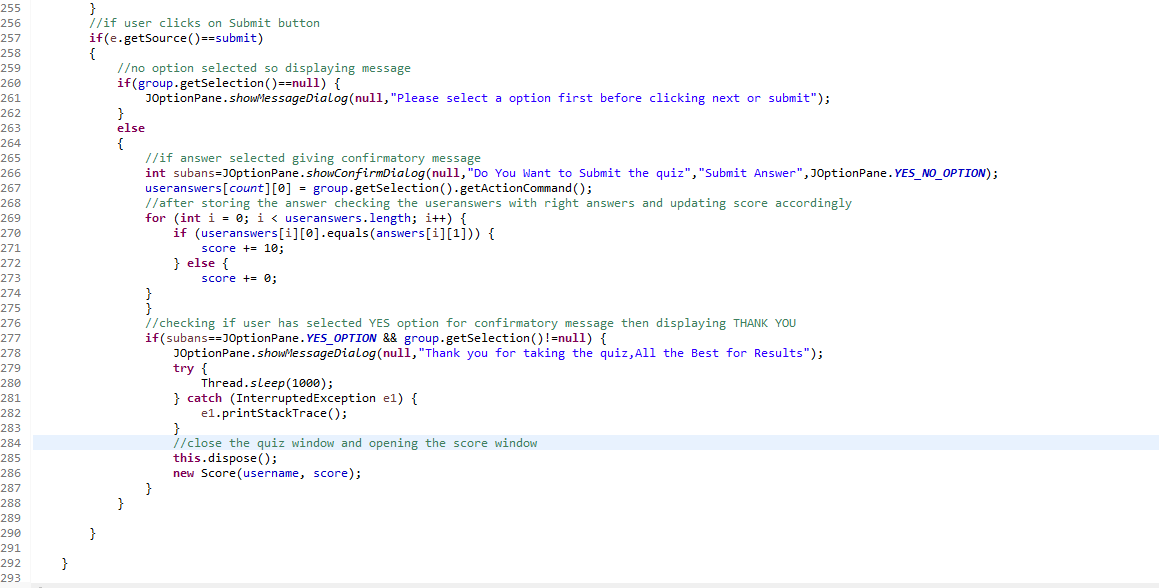
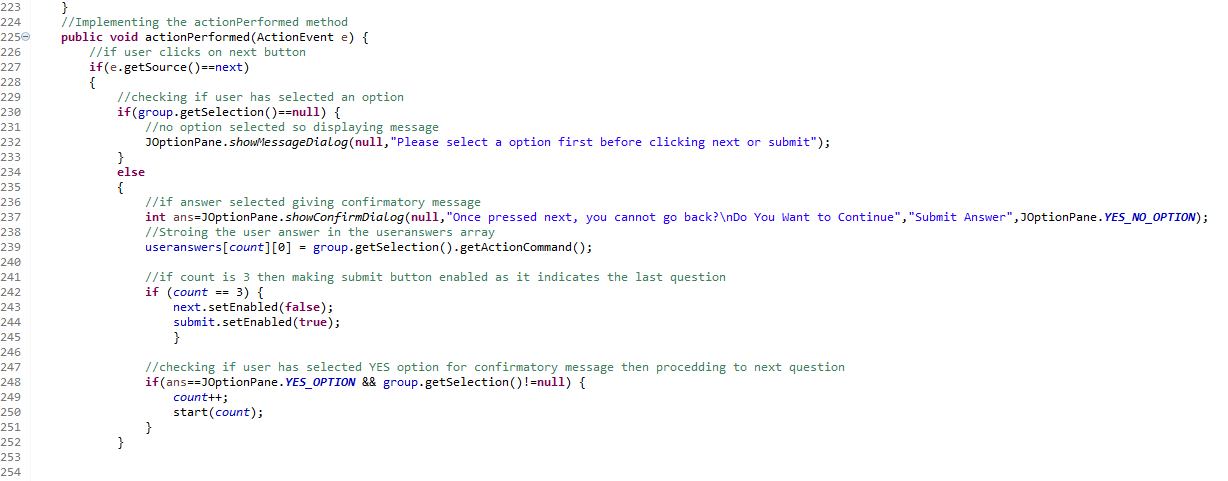
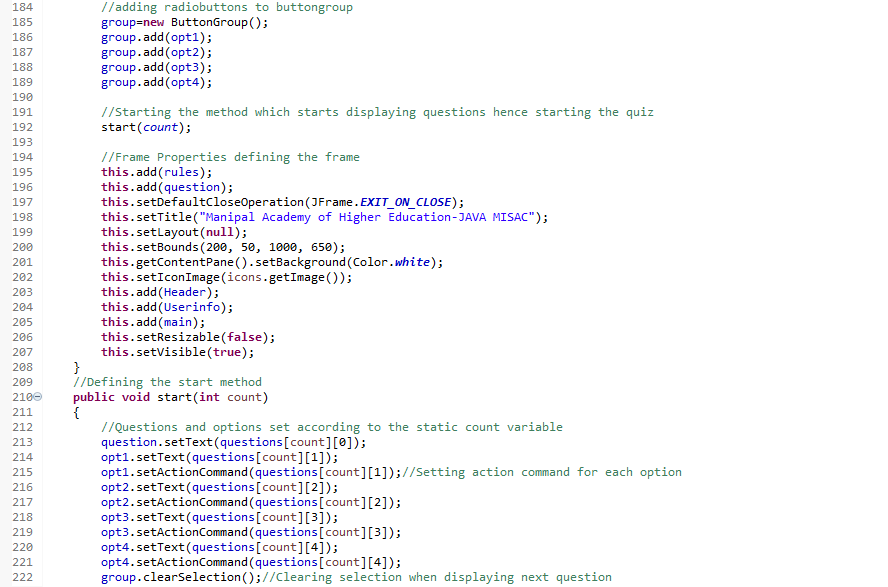
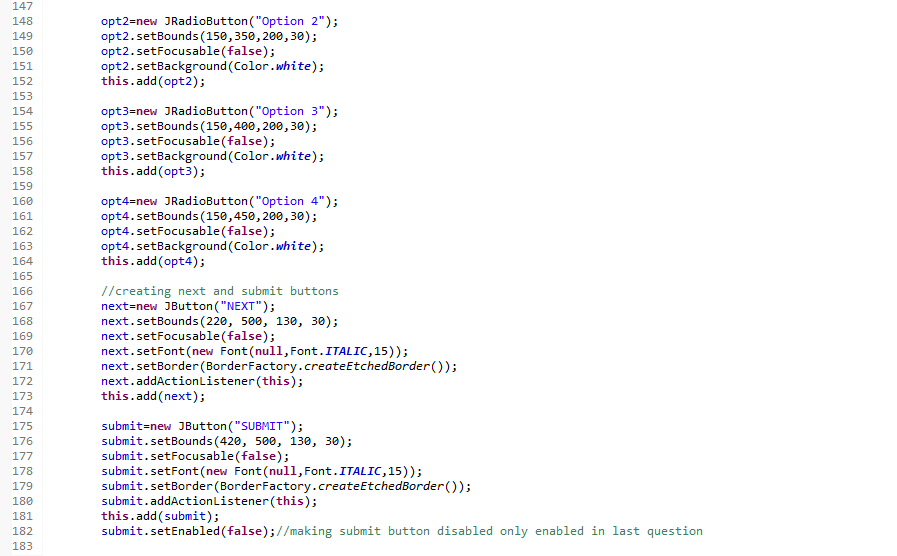
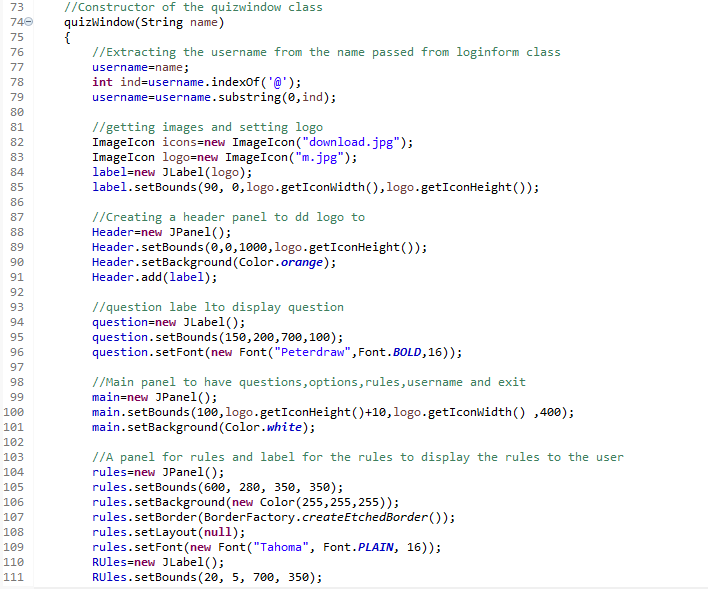
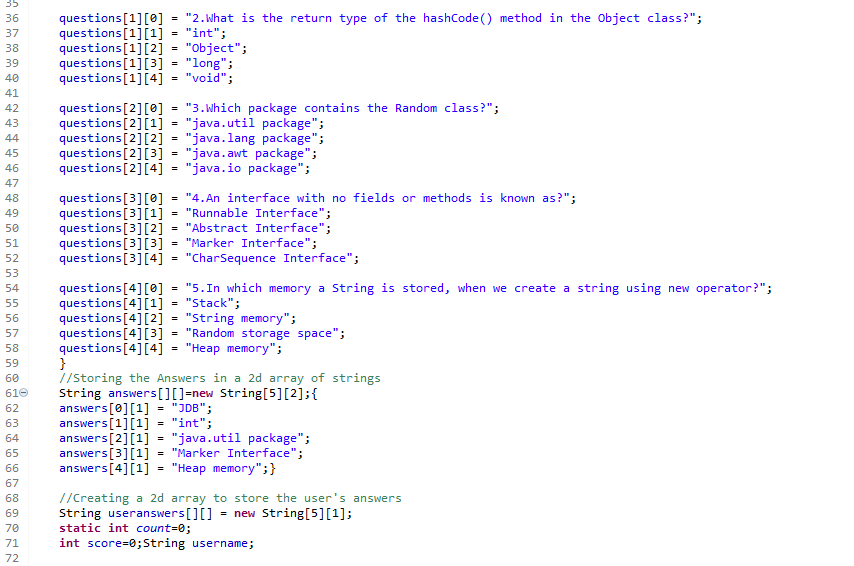
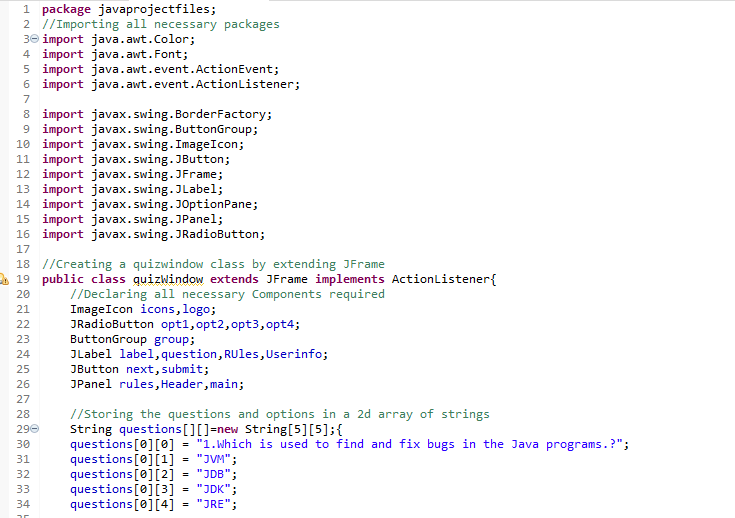
***1. LoginForm Class:***

[***https://drive.google.com/file/d/16jTOZB1Ag9rDOFOj-DO7UEEf85lI0f21/view?usp=sharing***](https://drive.google.com/file/d/16jTOZB1Ag9rDOFOj-DO7UEEf85lI0f21/view?usp=sharing)

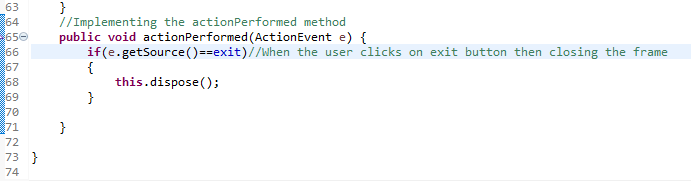
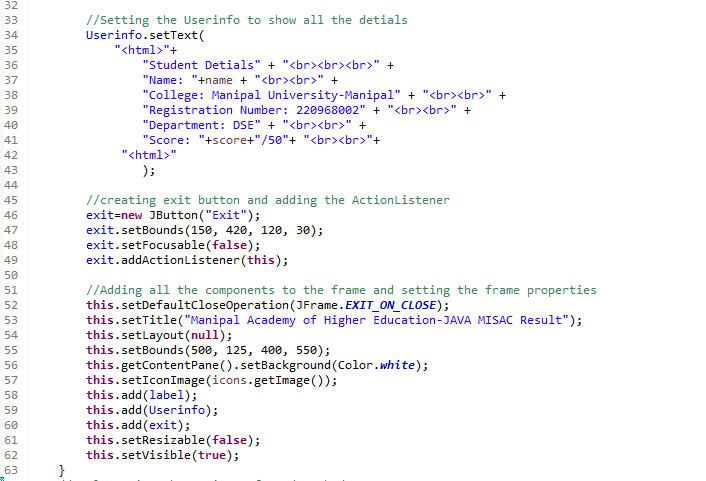
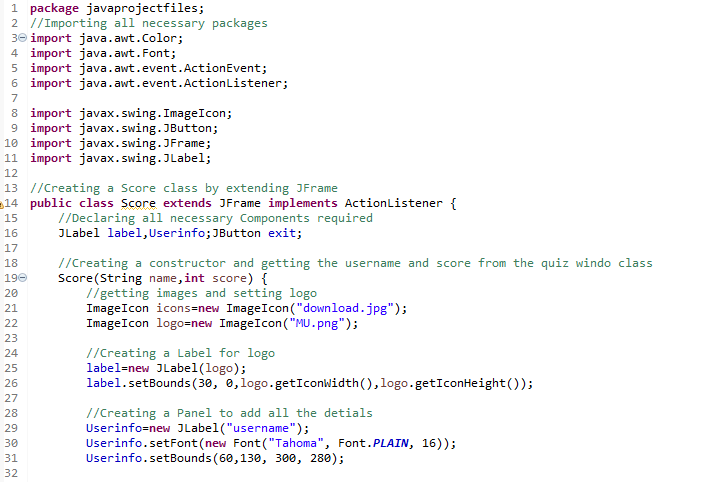


***2. QuizWindow Class:***

[***https://drive.google.com/file/d/1sEE2zdVeEDCEMtaEn-CpIIgvaBampGdh/view?usp=sharing***](https://drive.google.com/file/d/1sEE2zdVeEDCEMtaEn-CpIIgvaBampGdh/view?usp=sharing)

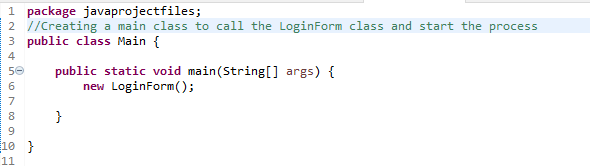


***3. Score Class:***

[***https://drive.google.com/file/d/16LdY-9umQGxPldsZPbCsovPxko6a4\_PE/view?usp=sharing***](https://drive.google.com/file/d/16LdY-9umQGxPldsZPbCsovPxko6a4_PE/view?usp=sharing)

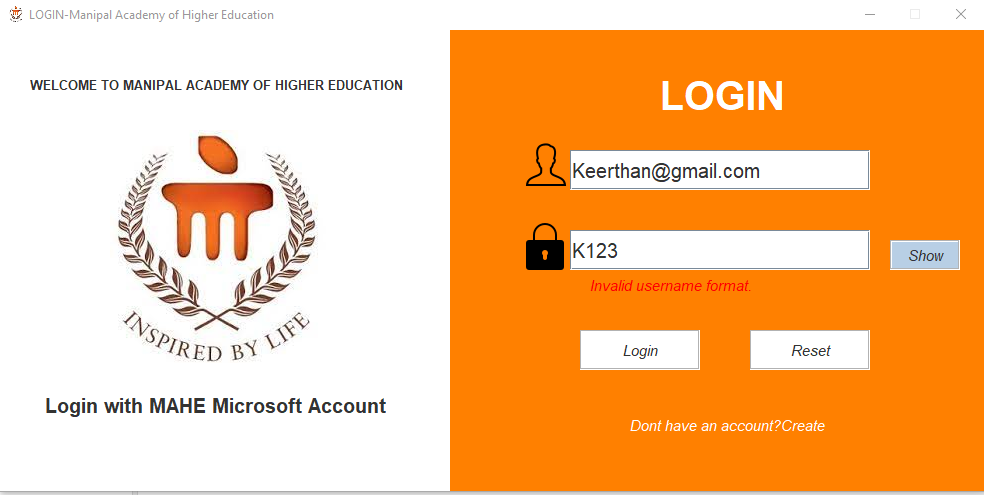
***4. Main Class:***

[***https://drive.google.com/file/d/1M3BeEXVaQfIJ3a4FLJ9CxAJPSGJjvCtc/view?usp=sharing***](https://drive.google.com/file/d/1M3BeEXVaQfIJ3a4FLJ9CxAJPSGJjvCtc/view?usp=sharing)

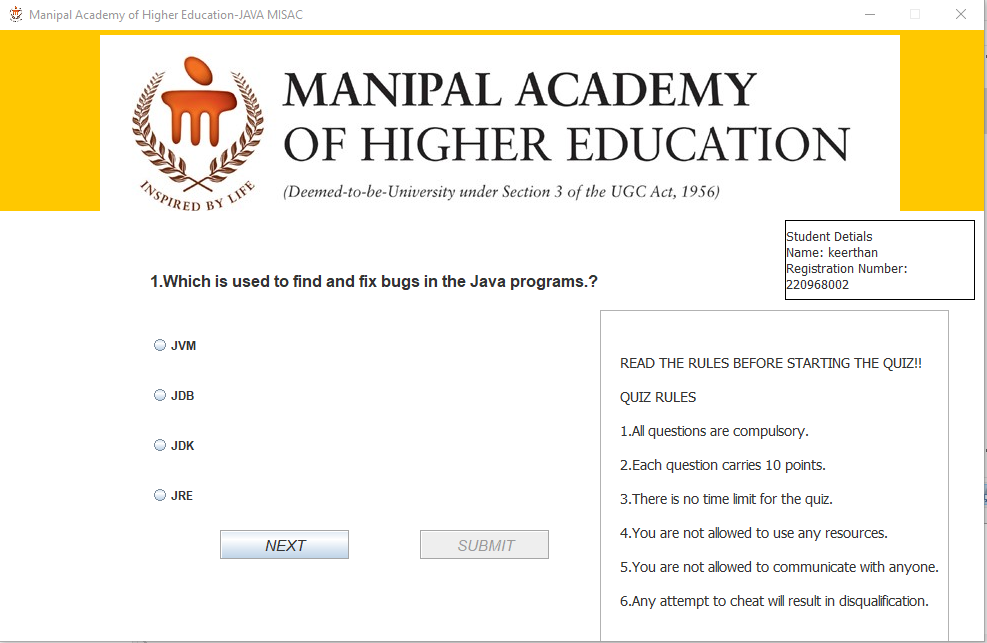
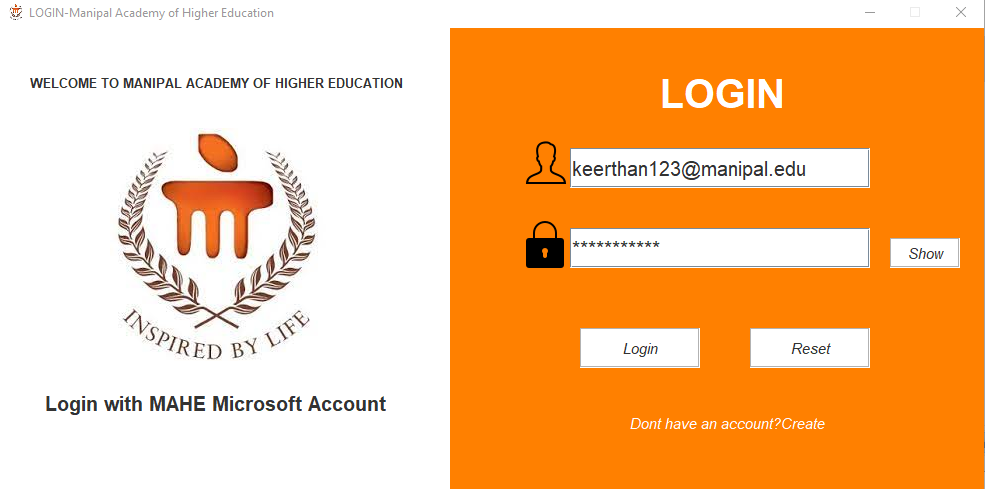


# Screenshots of outputs:

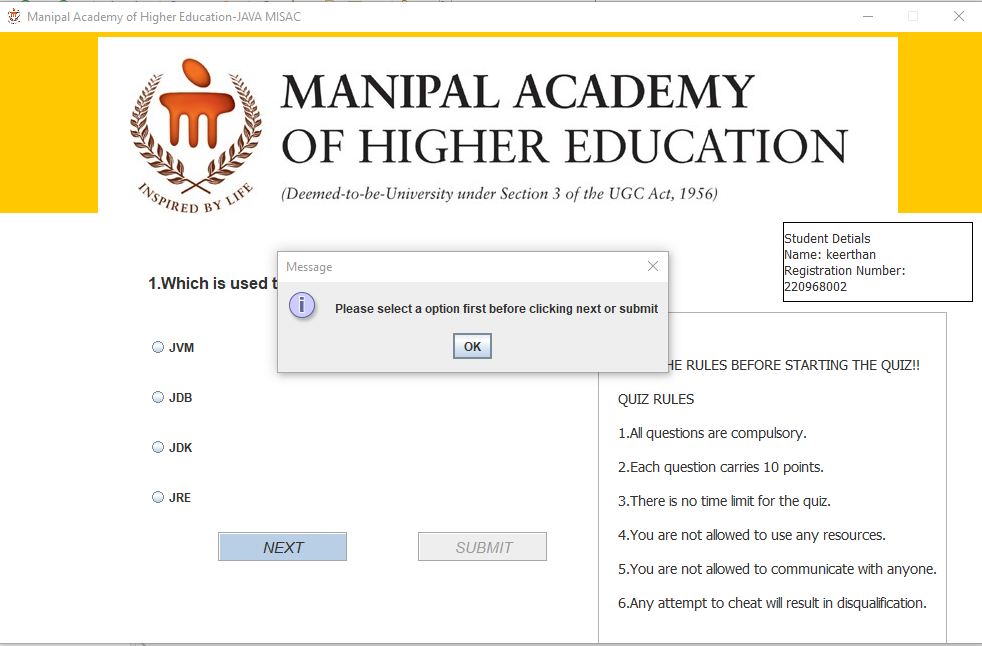
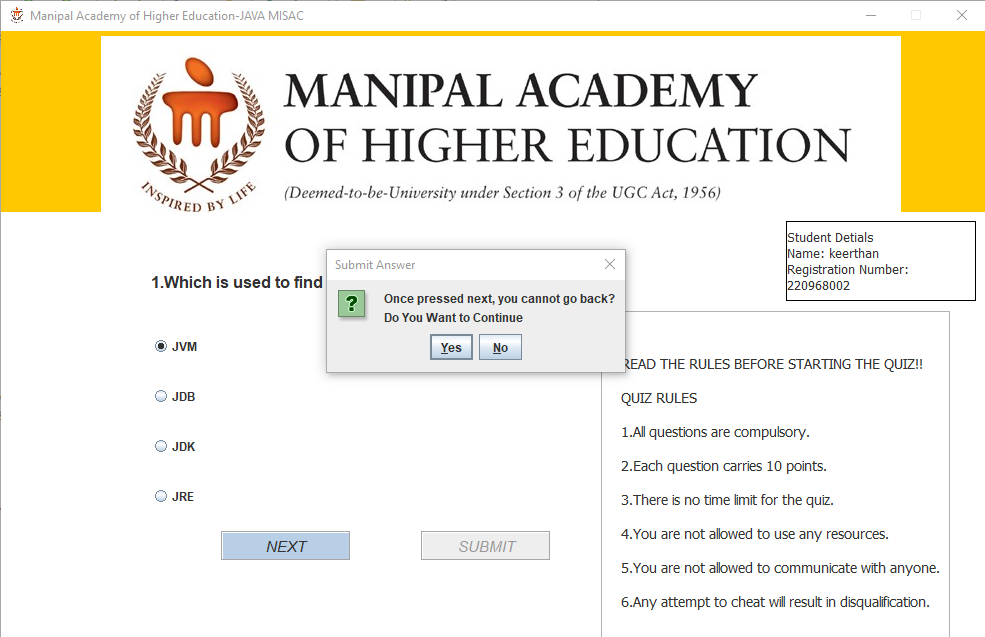
1. Invalid Username and Password showing error and password shown on clicking show button.



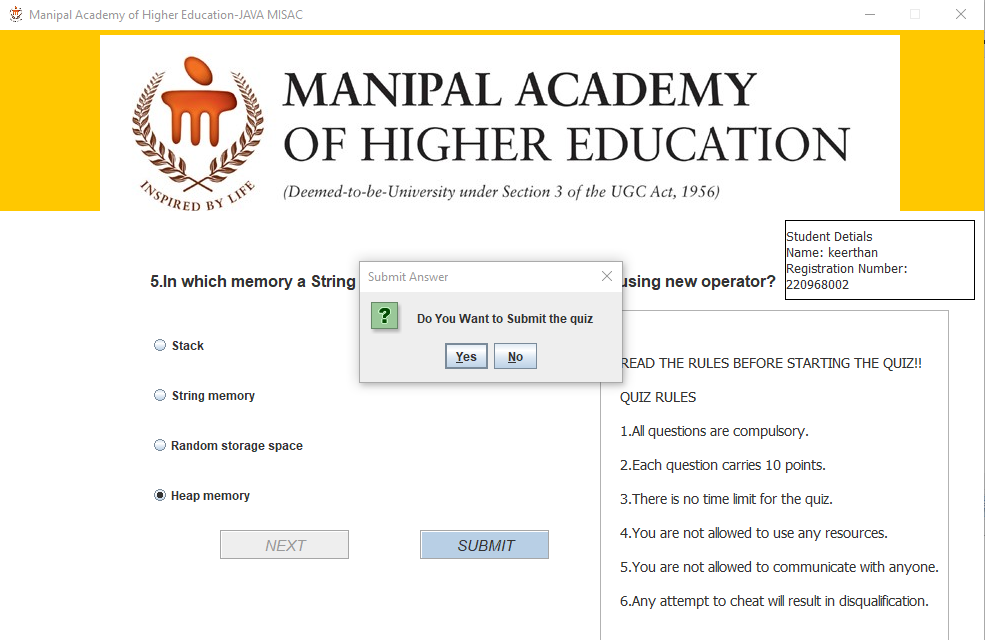
2. Proper format, on clicking the login go to the quiz window.

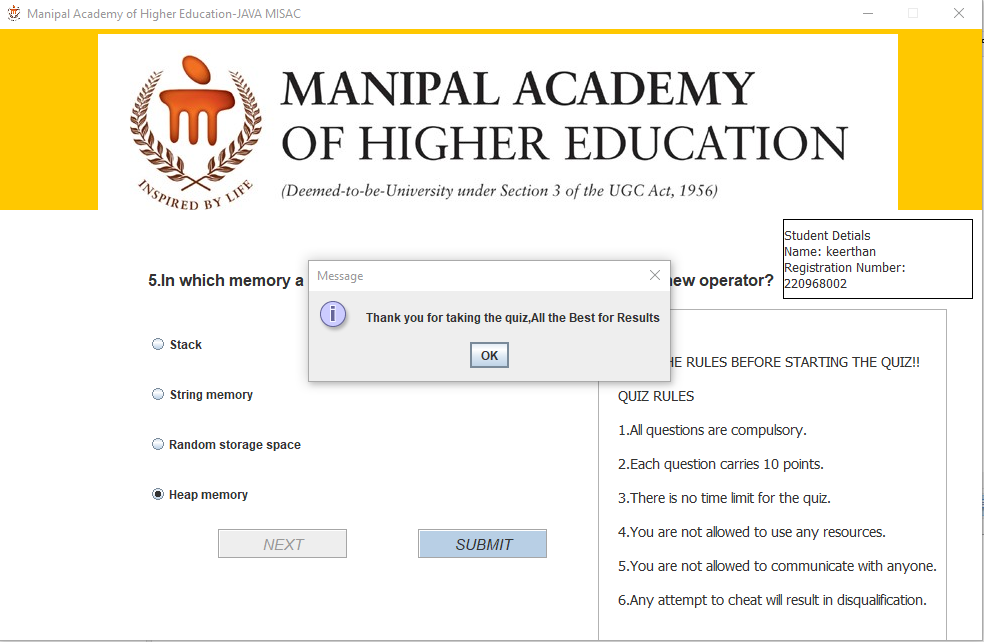


3. When no option is selected, and clicked on next shows a pop-up to select the option first.

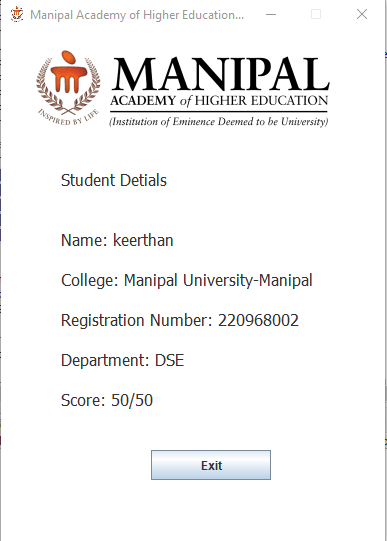
3. 4. When the option is selected and clicked on next shows a pop-up to Confirm

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>