



Lab 5: GUI- Menu Bar- Inputs / Outputs

Exercise 1:

Problem Description

Design a simple text editor in Java using Swing that allows users to type text in a text area and save it to a file or open an existing file. The editor should use a JMenuBar with menus and menu items for handling file operations.

Requirements

1. Graphical User Interface:

- Use JMenuBar, JMenu, and JMenuItem for the menu system.
- Include the following menu items under the **File** menu:
 - **New**: Clears the text area.
 - **Open**: Opens a file and loads its content into the text area.
 - **Save**: Saves the current text to a file.
 - **Exit**: Closes the application.

2. Text Area:

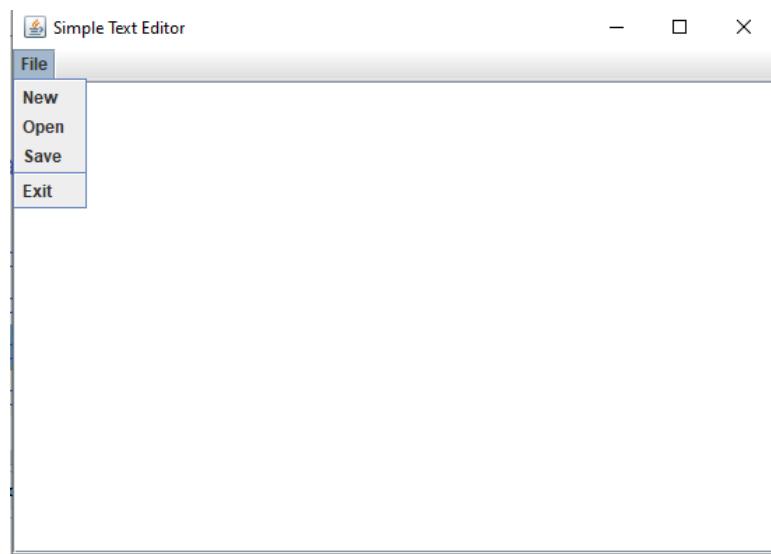
- Provide a JTextArea for the user to input and edit text.
- Use a scrollable pane (JScrollPane) for the text area.

3. File Handling:

- Use Java's FileWriter and FileReader for saving and reading files.
- Display file dialogs (JFileChooser) for file selection.

4. Error Handling:

- Handle file I/O exceptions gracefully with error messages.



Exercise 2: Employee Management System Using Swing

Problem Description

Create a Java Swing application that allows managing employee records in a single file. The application should have two modes:

1. Add Employee:

- A form to input employee details: **Name**, **ID**, **Department**, and **Salary**.
- A "Save" button to store the entered data in a text file.

2. Search Employee:

- A form with a search field to input an employee's **ID**.
- A "Search" button to retrieve and display the employee's information if found. Otherwise, display a message stating "Employee not found."



Add Employee

Employee ID:	<input type="text"/>
Name:	<input type="text"/>
Department:	<input type="text"/>
Salary:	<input type="text"/>

Employee Management System

Add Employee **Search Employee**

Save **Back**

Search Employee

Enter Employee ID: **Search**

Back