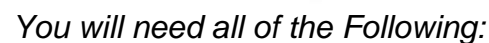


**If you do not complete this exercise during the lab period, you need to complete the work outside of the lab period and bring the completed work to the lab next week.**

The purpose of this exercise is to create an interactive Java GUI application, which requires the use of a separate class to handle the events of the menu items.

This application will behave similar fashion to the previous practice exercise except that there are no buttons and the options are given in the menu as shown below. You will attempt in making a note pad application.



- 1- JFrame
- 1- JMenuBar
- 1- JMenu
- 4- JMenuItem
- 1- JTextArea
- 1- JScrollPane

- Step 1: Create a text area
  - Create a class name Note Pad which. Create a text area of 20 rows and 15 columns. Using the default BorderLayout manager for a JFrame, put this text area in the center area.
- Step 2: Create a menu bar
  - Create a menu bar and add it to the frame.
- Step 3: Create a menu
  - Create one menu named **File** as shown above.
  - Compile and run your program to see that frame appears as above. At this point, the menu will not respond to a left click.
- Step 4: Add menu items to the menu
  - Add four menu items [bold part only] to the menu in the following order:
    - **Save** – Save the file
    - **Load** – Load to load a file
    - **Reset** – Clear the page
    - **Exit** – terminate the program.
- Step 5: Create a separate listener class named **handler** What data items must be passed to the constructor of this class? (Hint: how do you add to the Text Area)
- Step 6: Instantiate the **handler** object in MyMenu Write the statements to add this new object as the listener for the four menu items.
- Step 7: Implementing the actionPerformed method in **handler**
- Step 8: Writing the code for the Save/Load/reset/exit menu items The action of these menu item is simple and straight forward.

## Part 2:

This application will teach you how to handle mouse events, you will create a frame that only contains 1 JPanel and 1 JLabel, the label will be used to show the status of the events (e.g. if the mouse is moving the label will say "Mouse is moving!!"). the panel will be used as a testing component where you will be asked to change some of its property when you do some action with your mouse.

- **Step 1:** create a class that contains the following:
  - JFrame
  - JLabel
  - JPanel
- **Step2:** initialize the components and do all the necessary steps for the frame to be visible (as we did in part 1).
- **Step3:** create another class called mouseHandler, this class should implement 2 interfaces (MouseListener, MouseMotionListener).
- **Step4:** implements the 7 unimplemented methods in the mouse listener class and make them do as the following:
  - When mouse is clicked: change the status label to "Mouse is clicked!!"
  - When mouse is entered the area: change the status label to "You have entered the area" and change the panel background to RED.
  - When mouse leaves the area: change the status label to "You have left the area" and change the panel background back to white.
  - When mouse is pressed: change the status label to "You pressed the mouse"
  - When mouse is released: change the status label to "You released the mouse"
  - When mouse is dragged: change the status label to "You are dragging the mouse"
  - When mouse is moving: change the status label to "Mouse is moving !!"
- **Step5:** pair the mouseHandler class with the panel.

