This lab is about building Graphical User Interfaces/ event programming in Java.

Note: For accessibility, add a text "tool tip" to every component (button etc) in your screen. Do this using the `setToolTipText("..")` method.

Part 1 –Develop a basic screen 0.2

As coded in class, create a screen class (that extends JFrame) to get a simple screen running that contains **one button.**

To "run" or use the screen: Use a main method. From the main method, instantiate your screen class.

Change these things and rerun your screen to see the impact:

- The setSize() parameters;
- comment out the setVisible(true);

Part 2 – Add GUI components 0.4

Add the following components to it:
Another button,
A label ,
+ two more GUI components of your choice

See a list of all the GUI component classes (the ones beginning with "J") in the Javax.Swing package from the Java API to give you ideas and let you know the class names to use
https://docs.oracle.com/javase/8/docs/api/javax/swing/package-summary.html

Use a **layout** (either BorderLayout or FlowLayout) to control where your GUI components appear.

Run your screen again to make the new components appear.

Part 3 – Implement "event programming" to make the GUI responsive 0.6

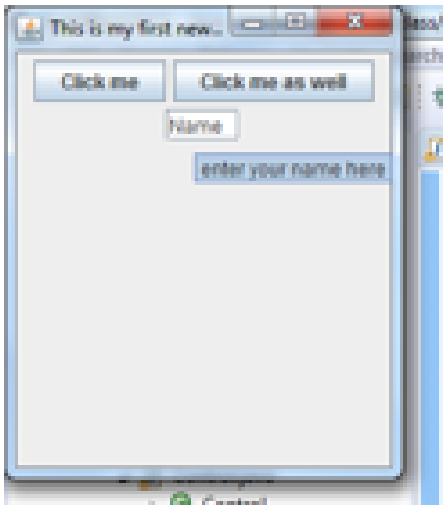
Add functionality so that when the first button is clicked, it displays a pop up method “first button clicked”.

Tip: display popup messages, the `JOptionPane` class has a static method: `showMessageDialog (this, "whatever your message is")` as demo’d in class.

Part 4 – Add more functionality to your screen 0.8

Make the second button clickable as well to display “second button clicked”. Make sure both buttons are still working correctly .

Add a JTextField as shown in the diagram below, with default text “Name” in it. Add “event programming” to the JTextField so that when you enter in text to the JTextField , a popup is displayed with the name you entered. (this event is captured by the same listener as buttons, `ActionListener`).
(hint: look at “getText” method of the JTextField class).

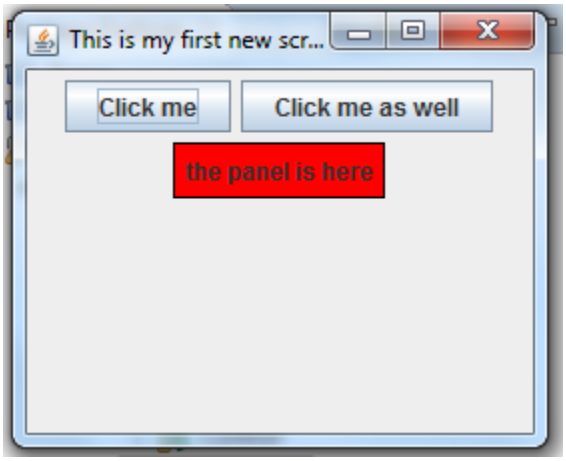


Part 5 – capture “mouse” events – 1

“MouseListener” is used to capture mouse events (e.g. mouse pressed, mouse clicked etc). Don’t forget you can implement more than one interface in a class.

Add the following functionality , which is also shown in the picture:

Add a panel (JPanel class) in to your screen.
Create a JLabel with text in it, and add that label to your panel as shown below.
Set the background colour of the panel to red:



Add mouse event detection so that:

- When the mouse *enters* the panel, it pops up a message “Mouse entered the panel”;
- When the mouse *exits* the panel, it pops up a message “Mouse left the panel”;
- When you click on the panel, it pops up a message to say clicked. See if you can get it to distinguish between “right” and “left” mouse clicks.