Simple Guide to Simple Swing Building & Event Handling in Java
---
An 'event' is an object that acts as a signal to another object
known as a 'listener'. GUI components generate/send/fire events
A listener object has methods that specifies the action(s) to
be when events of various kinds are received by the listener.
Methods that handle events are called 'event handlers'. For each
GUI component needs to specify what objects are listeners that
will respond to events generated/sends/fired by the GUI, this
is called 'registering a listener'.
    Different components generate different kinds of events. A
JButton components generates 'action events', which are handled
by action listeners. An action listener is an object whose class
implements the ActionListener interface. The ActionListener
interface has only one method heading that must be implemented:

```
    public void actionPerformed(ActionEvent e)
```

where e above can be any valid identifier.

    +++ To Create a Swing GUI +++
1. Create a class that extends JFrame;
2. import java.awt.*;   // For Layout Managers
   import javax.swing.*;   // For Swing GUI Components
   import java.awt.event.*; // For ActionListener & ActionListener
3. Declare all components used in the GUI (including JPanels).

    +++ Inside the class constructor +++
4. Initialise all components inside the class constructor.
5. add GUI components to appropiate JPanels.
6. add appropiate JPanels to JFrame.
7. Declare listeners and register them with components that use them.

    +++ To Test a Swing GUI +++
======= should be inside main function=======
8. import javax.swing.JFrame (if main is in another class)
9. Create an object (say m) of the above class;
10. m.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11. m.setSize(x, y);
12. m.setVisible(true);

    +++ To Handle Event Handling +++
13. Create a private event handler class that implements ActionListener
14. Inside the handler class implement a function
        public void actionPerformed(ActionEvent event)
   that handles action(s) to be performed.

15. Test the program
Note: Order of steps in coding: 2,1,3,4,5,6,7,13,14,15,9,10,11,12.
Note: The order above is not unique.