Gradual Development of a Swing Application

(10 Steps)

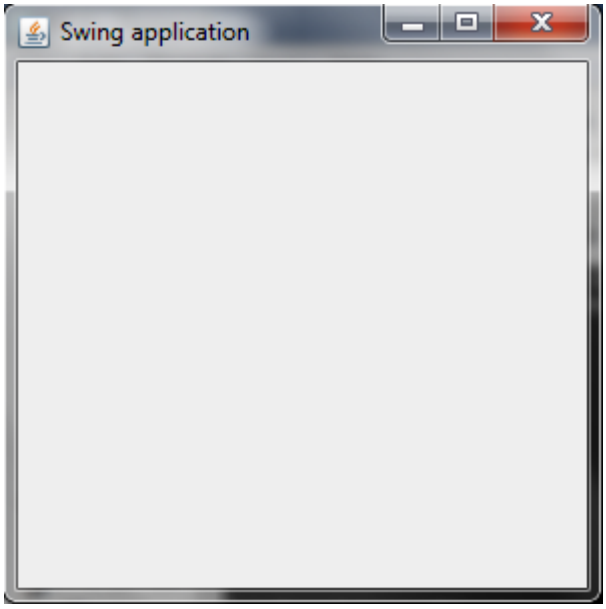
I. Write the following code

```
1. import java.awt.*;
2. import javax.swing.*;
3. class MSA {
4. public static void main(String args[]){
5. JFrame jf=new JFrame("Swing application");
6. jf.setVisible(true);
7. }
8. }
```

This produces the following output.

II. Add a line of code, jf.setSize(300,300); after line 5.

```
1. import java.awt.*;
2. import javax.swing.*;
3. class MSA {
4. public static void main(String args[]){
5. JFrame jf=new JFrame("Swing application");
6. jf.setSize(300,300);
7. jf.setVisible(true);
8. }
9. }
```



III. By this time, you might have observed that the swing application is not closing by closing JFrame window. To achieve this, add a statement, jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

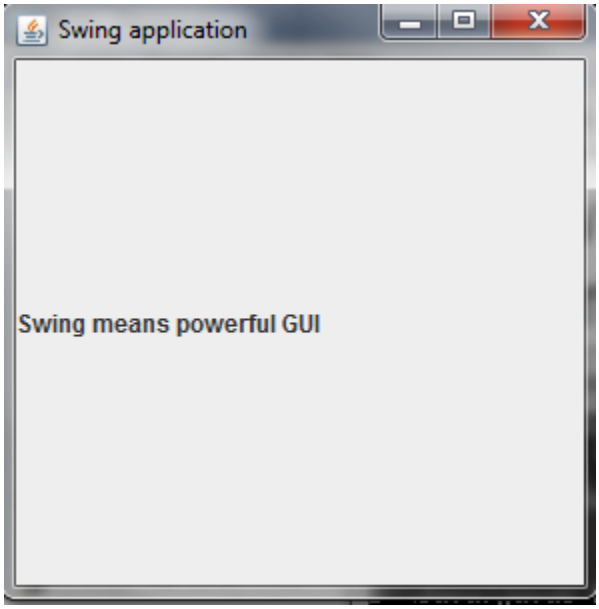
```
1. import java.awt.*;
2. import javax.swing.*;
3. class MSA {
4. public static void main(String args[]){
5. JFrame jf=new JFrame("Swing application");
6. jf.setSize(300,300);
7. jf.setVisible(true);
8. jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
9. }
10. }
```

You can observe, when a JFrame window is closed, Java application also terminates its execution.

IV. Display some text on JFrame: This can be done by adding a label. Create a JLabel object and add it to jf (lines 8 and 9).

```
1. import java.awt.*;
2. import javax.swing.*;
3. class MSA {
4. public static void main(String args[]){
```

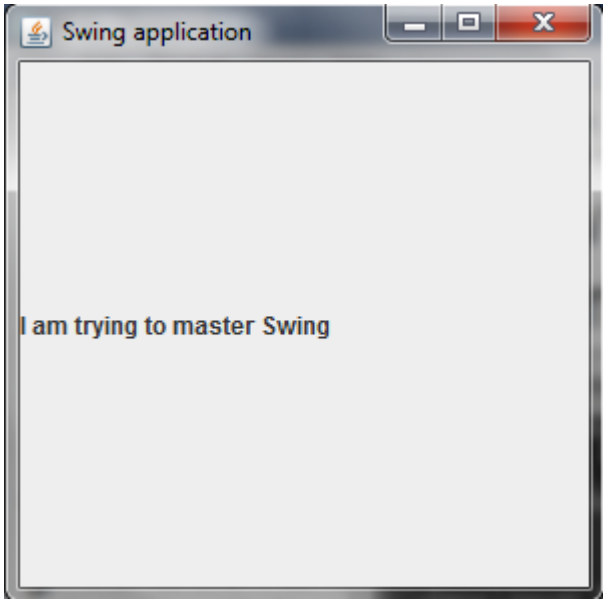
5. JFrame jf=new JFrame("Swing application");
6. jf.setSize(300,300);
7. JLabel l=new JLabel("Swing means powerful GUI");
8. jf.add(l);
9. jf.setVisible(true);
10. jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11. }
12. }



V. Add one more label (lines 9 and 10)

1. import java.awt.*;
2. import javax.swing.*;
3. class MSA {
4. public static void main(String args[]){
5. JFrame jf=new JFrame("Swing application");
6. jf.setSize(300,300);
7. JLabel l=new JLabel("Swing means powerful GUI");
8. jf.add(l);
9. JLabel l1=new JLabel("I am trying to master Swing");
10. jf.add(l1);
11. jf.setVisible(true);
12. jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
13. }
14. }

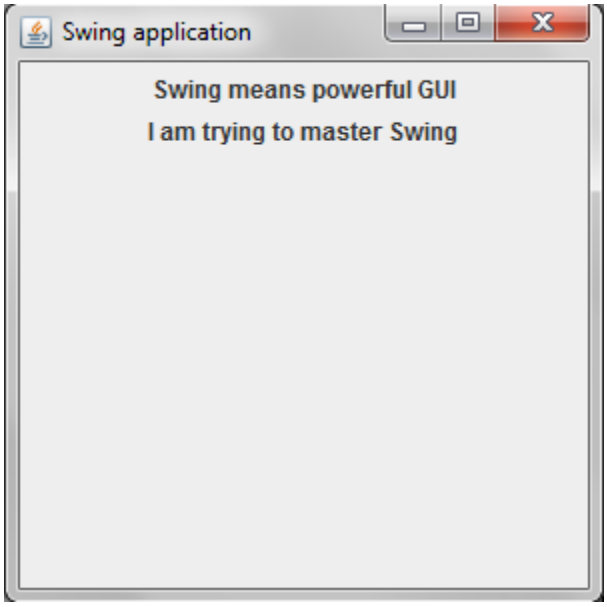
You would expect to display two labels. But only the recently added label on the window.



VI. To avoid this overlap of labels, set Layout to FlowLayout(line 6)

```
1. import java.awt.*;
2. import javax.swing.*;
3. class MSA {
4. public static void main(String args[]){
5. JFrame jf=new JFrame("Swing application");
6. jf.setLayout(new FlowLayout());
7. jf.setSize(300,300);
8. JLabel l=new JLabel("Swing means powerful GUI");
9. jf.add(l);
10. JLabel l1=new JLabel("I am trying to master Swing");
11. jf.add(l1);
12. jf.setVisible(true);
13. jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
14. }
15. }
```

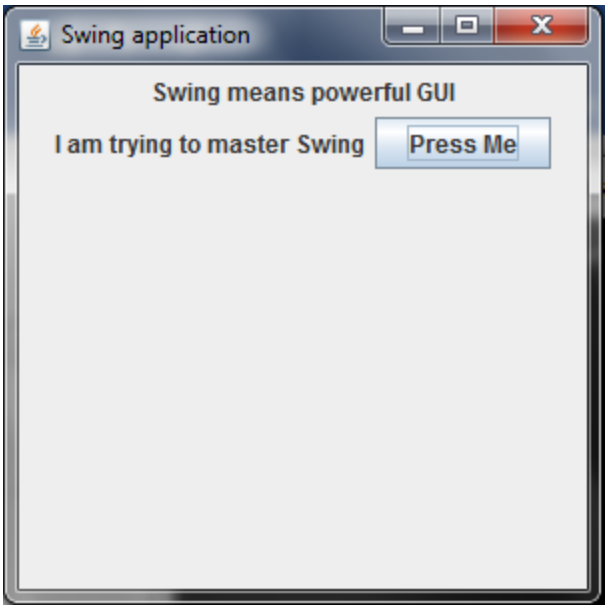
The output is like this.



VII. Add a button (lines 12 and 13)

```
1. import java.awt.*;
2. import javax.swing.*;
3. class MSA {
4. public static void main(String args[]){
5. JFrame jf=new JFrame("Swing application");
6. jf.setLayout(new FlowLayout());
7. jf.setSize(300,300);
8. JLabel l=new JLabel("Swing means powerful GUI");
9. jf.add(l);
10. JLabel l1=new JLabel("I am trying to master Swing");
11. jf.add(l1);
12. JButton b1=new JButton("Press Me");
13. jf.add(b1);
14. jf.setVisible(true);
15. jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16. }
17. }
```

The output looks like this.



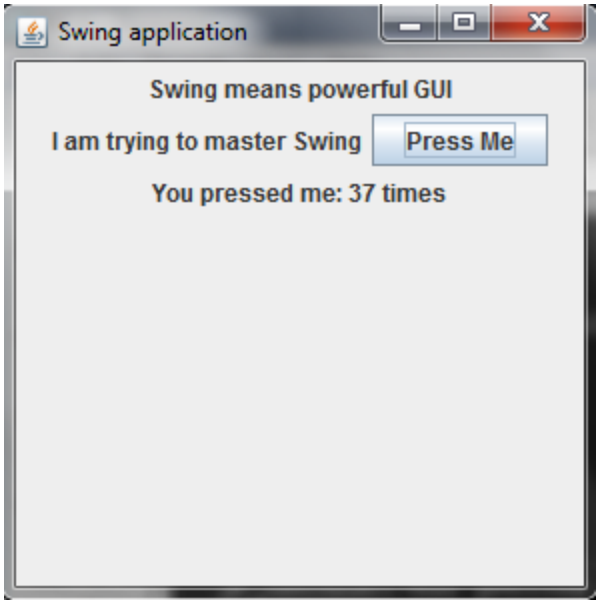
VIII. We can press on the button. This is called an event. But, nothing would happen when the button is pressed. Making something happen when an event has occurred is called event handling. We can do this by ActionListener (lines 2, 16, 17, 18, 19, 20, 21, 22, and 23)

```
1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4. class MSA {
5. static int count=0;
6. public static void main(String args[]){
7. JFrame jf=new JFrame("Swing application");
8. jf.setLayout(new FlowLayout());
9. jf.setSize(300,300);
10. JLabel l=new JLabel("Swing means powerful GUI");
11. jf.add(l);
12. JLabel l1=new JLabel("I am trying to master Swing");
13. jf.add(l1);
14. JButton b1=new JButton("Press Me");
15. jf.add(b1);
16. JLabel l2=new JLabel();
17. jf.add(l2);
18. b1.addActionListener(new ActionListener()
19. {public void actionPerformed(ActionEvent ae){
20. l2.setText("You pressed me: "+ ++count +" times");
```

21. }
22. }
23.);
24. jf.setVisible(true);
25. jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

}

26. }

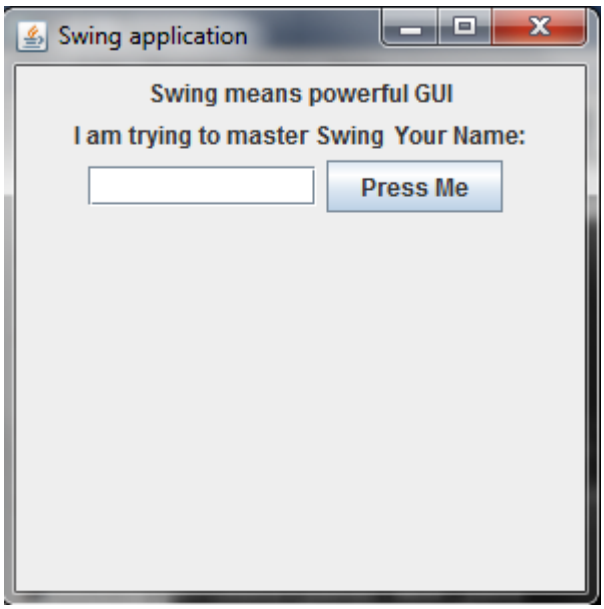


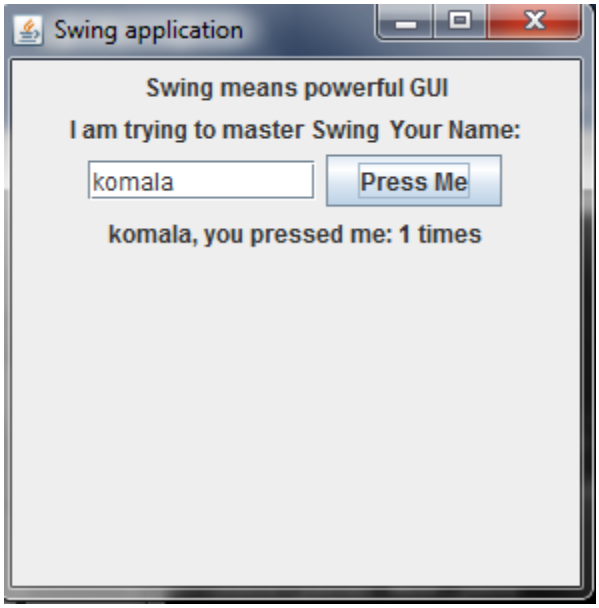
IX. Adding a text box to JFrame (lines 16, 17, and 24)

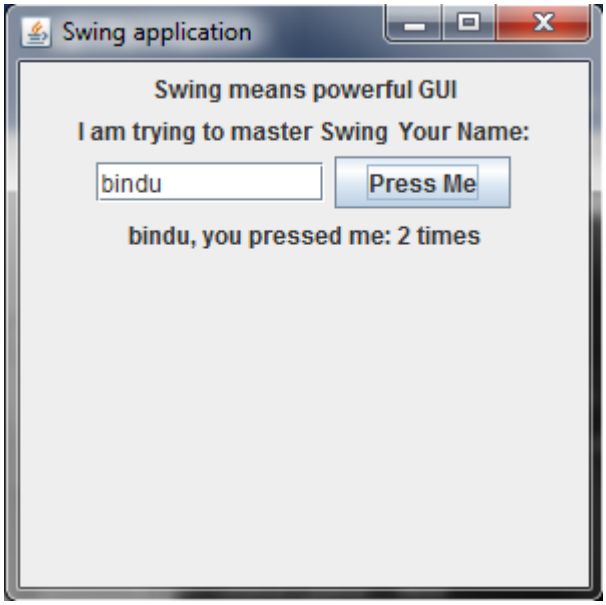
```
1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4. class MSA {
5. static int count=0;
6. public static void main(String args[]){
7. JFrame jf=new JFrame("Swing application");
8. jf.setLayout(new FlowLayout());
9. jf.setSize(300,300);
10. JLabel l=new JLabel("Swing means powerful GUI");
11. jf.add(l);
12. JLabel l1=new JLabel("I am trying to master Swing");
13. jf.add(l1);
14. JLabel nl=new JLabel("Your Name:");
15. jf.add(nl);
16. JTextField t=new JTextField(10);//width of 10 columns
17. jf.add(t);
```

```
18. JButton b1=new JButton("Press Me");
19. jf.add(b1);
20. JLabel l2=new JLabel();
21. jf.add(l2);
22. b1.addActionListener(new ActionListener()
23. {public void actionPerformed(ActionEvent ae){
24. l2.setText(t.getText()+", you pressed me: "+ ++count +" times");
25. }
26. }
27. );
28. jf.setVisible(true);
29. jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30. }
31. }
```

The output at various stages is shown below.







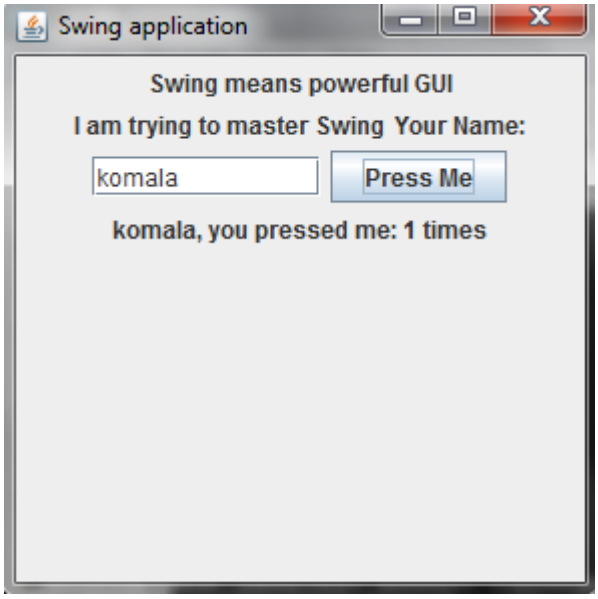
One observation here is, though 'bindu' pressed the button for the first time, we get the message, 'bindu, you pressed me: 2 times'. This is because the 'count' is not reset to 0.

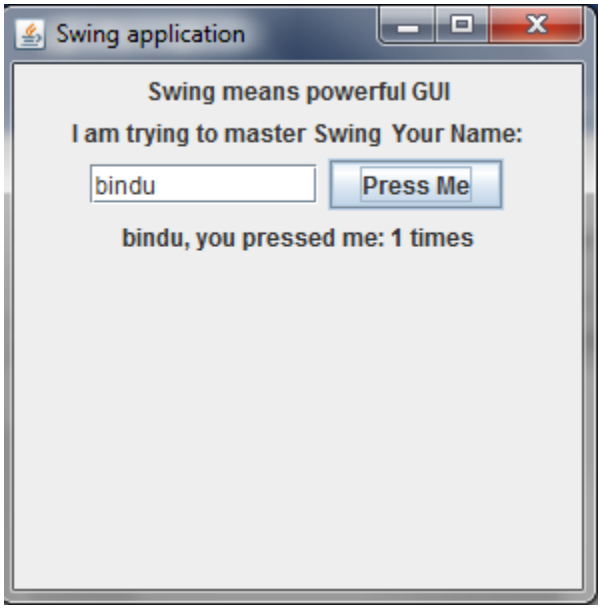
X. Reset count to 0 when the focus is in text box. To do this, we should add FocusListener to text field, instead of ActionListener (lines 22 to 28). ActionListener works when a return key is pressed.

1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;

```
4. class MSA {
5. static int count=0;
6. public static void main(String args[]){
7. JFrame jf=new JFrame("Swing application");
8. jf.setLayout(new FlowLayout());
9. jf.setSize(300,300);
10. JLabel l=new JLabel("Swing means powerful GUI");
11. jf.add(l);
12. JLabel l1=new JLabel("I am trying to master Swing");
13. jf.add(l1);
14. JLabel nl=new JLabel("Your Name:");
15. jf.add(nl);
16. JTextField t=new JTextField(10);//width of 10 columns
17. jf.add(t);
18. JButton b1=new JButton("Press Me");
19. jf.add(b1);
20. JLabel l2=new JLabel();
21. jf.add(l2);
22. t.addFocusListener(new FocusListener()
23. {public void focusGained(FocusEvent fe){
24. count=0;
25. }
26. public void focusLost(FocusEvent fe){}
27. }
28. );
29. b1.addActionListener(new ActionListener()
30. {public void actionPerformed(ActionEvent ae){
31. l2.setText(t.getText()+", you pressed me: "+ ++count +" times");
32. }
33. }
34. );
35. jf.setVisible(true);
36. jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
37. }
38. }
```

The output looks like this.


Swing application
Swing means powerful GUI
I am trying to master Swing Your Name:
komala
Press Me
komala, you pressed me: 1 times


Swing application
Swing means powerful GUI
I am trying to master Swing Your Name:
bindu
Press Me
bindu, you pressed me: 1 times