GAURAV KISHOR PATIL
ROLL NO: 54
DIV:2 BATCH:C

Experiment No. 16
Java AWT and Event Handling
Date of Performance:29/09/23
Date of Submission:05/10/23

Aim :- To implement GUI application using Java AWT and Event Handling

Objective:- Develop a simple GUI to display a message in the textfield on pressing a button.

Theory:-

The Graphics class is the abstract base class for all graphics contexts that allow an application to draw onto components that are realized on various devices, as well as onto off-screen images.

Java provides a readymade package named awt(abstract windowing toolkit) which contains

various classes to support GUI designing.

What is an Event?

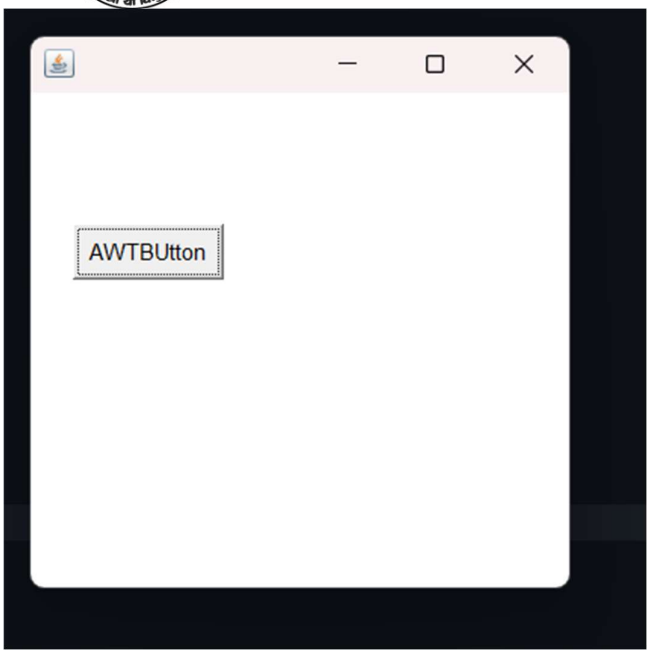
Change in the state of an object is known as event i.e. event describes the change in state of source. Events are generated as result of user interaction with the graphical user interface components. For example, clicking on a button, moving the mouse, entering a character through keyboard, selecting an item from list, scrolling the page are the activities that causes an event to happen.

What is Event Handling?
Event Handling is the mechanism that controls the event and decides what should happen if an event occurs. This mechanism have the code which is known as event handler that is executed when an event occurs. Java Uses the Delegation Event Model to handle the events. This model defines the standard mechanism to generate and handle the events.Let's have a brief introduction to this model.

Code :-

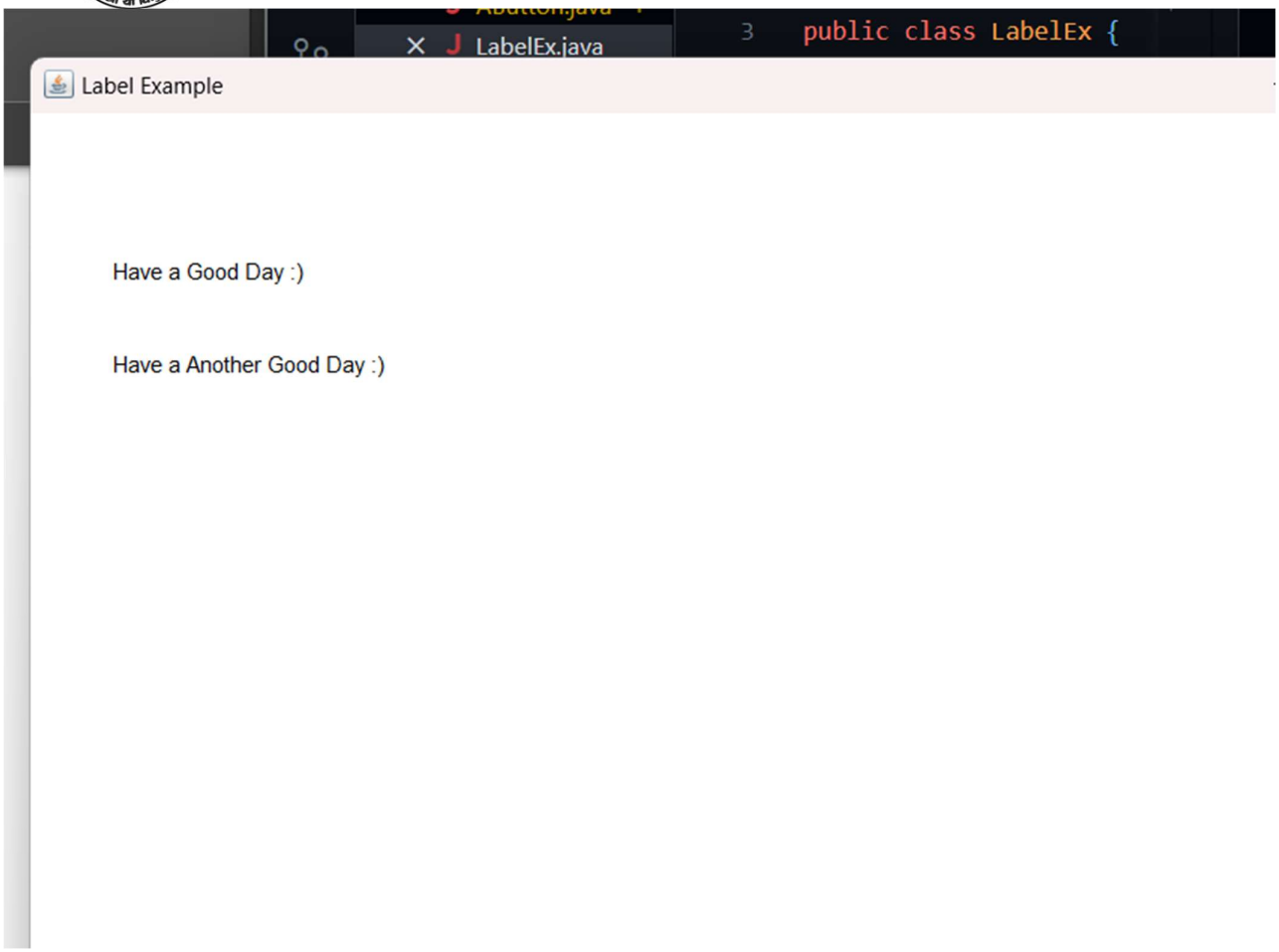
1) Buttons

```
import java.awt.*;

public class AButton extends Frame{

   AButton(){
   Button b = new Button("AWTBUtton");

   b.setBounds(30,100,80,30);
   add(b);
   setSize(300,300);
   setLayout(null);
   setVisible(true);


   }

   public static void main(String args[]){
      AButton f = new AButton();
   }

}
```



2)Label using awt

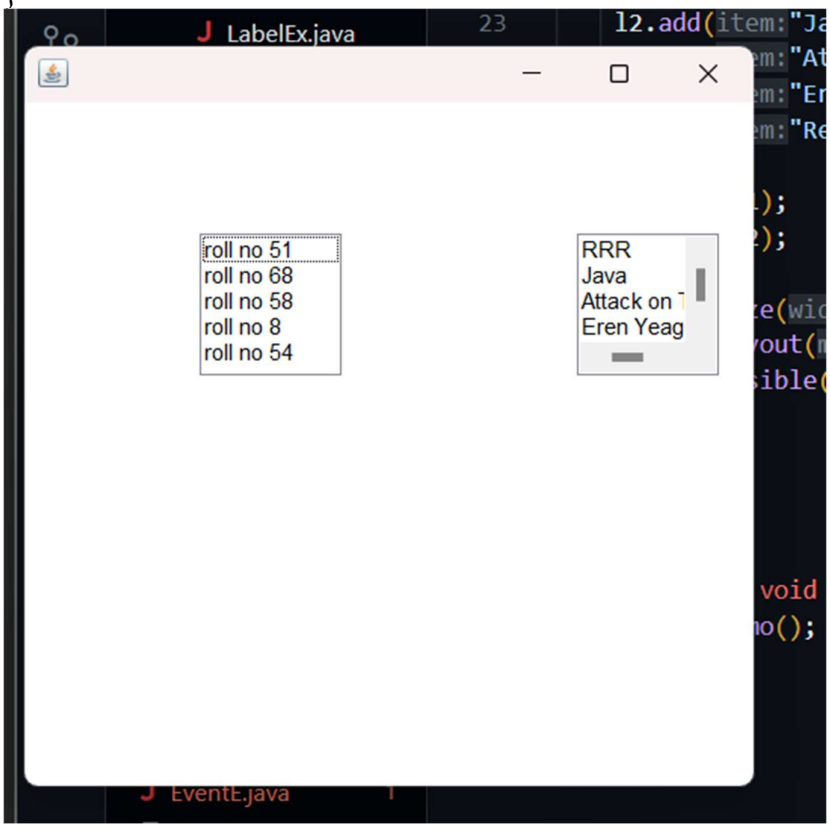
```
import java.awt.*;

public class LabelEx {
  public static void main(String args[]){
   Frame f = new Frame("Label Example ");

   Label l1 = new Label("Have a Good Day :)");
    Label l2 = new Label("Have a Another Good Day :)");

    l1.setBounds(50,100,300,30);
    l2.setBounds(50,150,300,30);

    f.add(l1);
    f.add(l2);

    f.setSize(800,800);

    f.setLayout(null);
    f.setVisible(true);


  }
}
```



3) List using awt

```
import java.awt.*;

public class ListDemo {


ListDemo(){


    Frame f = new Frame();

    List l1= new List(5);
    List l2= new List(5);

    l1.setBounds(100,100,75,75);
    l2.setBounds(300,100,75,75);

    l1.add("roll no 51");
    l1.add("roll no 68");
    l1.add("roll no 58");
    l1.add("roll no 8");
```

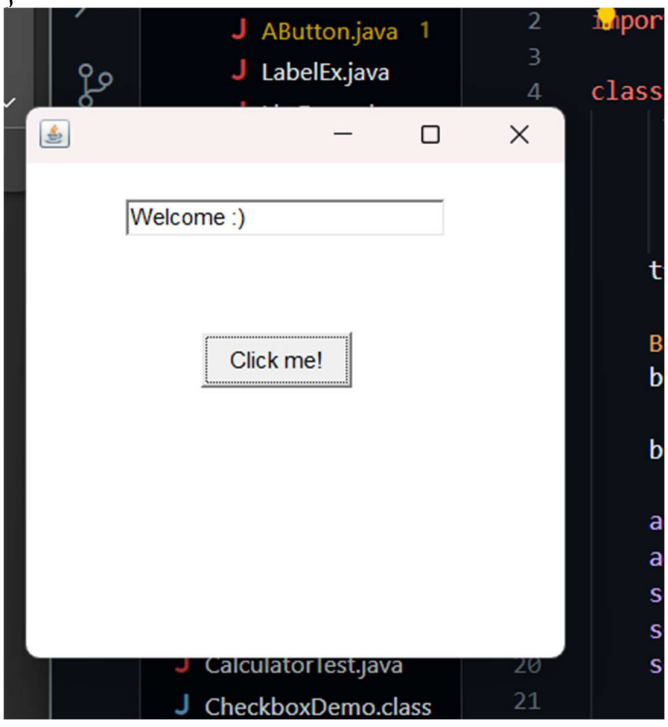
```
    l1.add("roll no 54");
    l2.add("RRR");
    l2.add("Java");
    l2.add("Attack on Titan");
    l2.add("Eren Yeager");
    l2.add("Red Dead");

      f.add(l1);
      f.add(l2);

      f.setSize(400,400);
      f.setLayout(null);
      f.setVisible(true);


}

public static void main(String args[]){
   new ListDemo();

}
}
```



4)Textfield using ActionListener

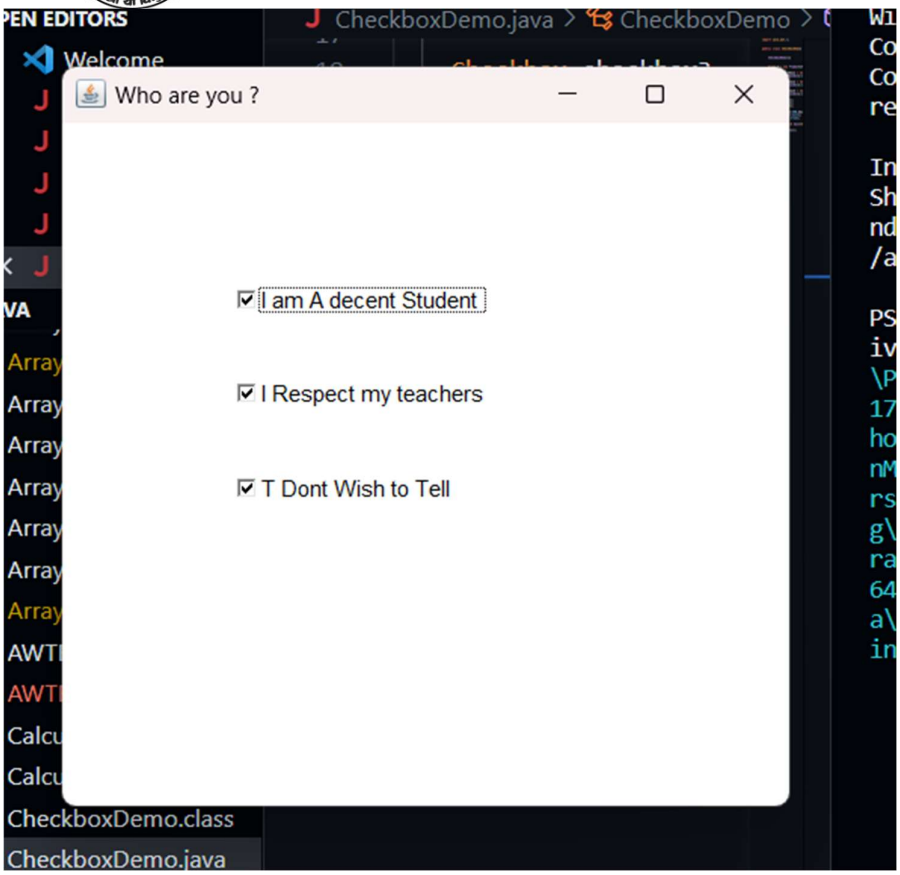
```
import java.awt.*;
import java.awt.event.*;
```

```
class EventE extends Frame implements ActionListener{
    TextField tf = new TextField();

     EventE(){

   tf.setBounds(60,50,170,20);

   Button b = new Button("Click me!");
   b.setBounds(100,120,80,30);

   b.addActionListener(this);

   add(b);
   add(tf);
   setSize(300,300);
   setLayout(null);
   setVisible(true);

}
public void actionPerformed(ActionEvent e){
   tf.setText("Welcome :)");
}

public static void main(String args[]){

   new EventE();
}
}
```



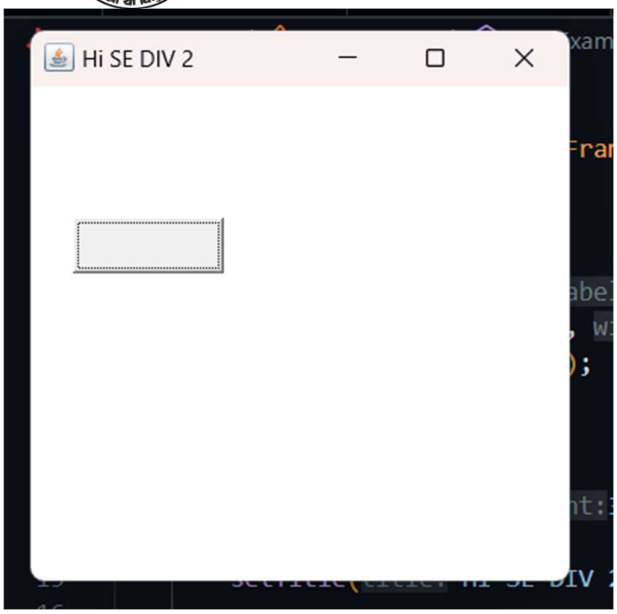
5)Checkbox

```
import java.awt.*;


public class CheckboxDemo{


   CheckboxDemo(){


   Frame f = new Frame("Who are you ?");

   Checkbox checkbox1 = new Checkbox("I am A decent Student ");
   checkbox1.setBounds(100,100,200,50);

   Checkbox checkbox2 = new Checkbox("I Respect my teachers",true);
   checkbox2.setBounds(100,150,200,50);

   Checkbox checkbox3 = new Checkbox("T Dont Wish to Tell",true);
   checkbox3.setBounds(100,200,200,50);

   f.add(checkbox1);
   f.add(checkbox2);
   f.add(checkbox3);

   f.setSize(400,400);
   f.setLayout(null);
   f.setVisible(true);
   }
  public static void main(String args[]){

   new CheckboxDemo();
  }

}
```



6) AWT button

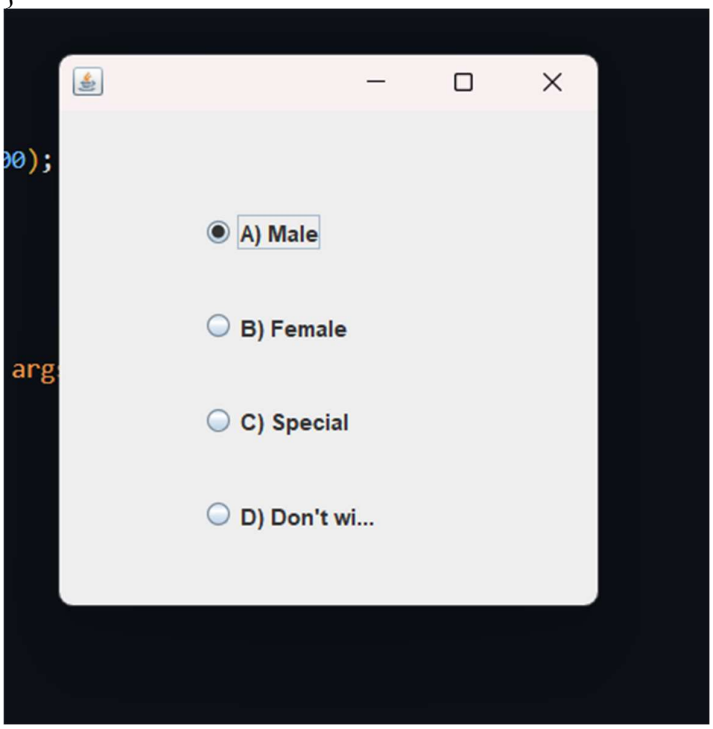
```
import java.awt.*;

public class AWTExamp1 extends Frame {

   AWTExamp1(){

      Button b = new Button("Click Me !");

      b.setBounds(30,100,80,30);

      add(b);

      setSize(300,300);

      setTitle("Hi SE DIV 2");

      setLayout(null);

      setVisible(true);
   }
   public static void main(String args[]){
      AWTExamp1 f = new AWTExamp1();
   }
}
```



7)RadioButton

```
import javax.swing.*;

public class RadioButtonEx{

    JFrame f = new JFrame();
    RadioButtonEx(){

        JRadioButton r1 = new JRadioButton("A) Male");
        JRadioButton r2= new JRadioButton("B) Female");
        JRadioButton r3= new JRadioButton("C) Special");
        JRadioButton r4 = new JRadioButton("D) Don't wish to tell");


        r1.setBounds(75,50,100,30);
        r2.setBounds(75,100,100,30);
        r3.setBounds(75,150,100,30);
        r4.setBounds(75,200,100,30);

        ButtonGroup bg = new ButtonGroup();
        bg.add(r1);
        bg.add(r2);
         bg.add(r3);
        bg.add(r4);
        f.add(r1);
        f.add(r2);
        f.add(r3);
        f.add(r4);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
```

```
    }

    public static void main(String[] args){

        new  RadioButtonEx();

    }
}
```



Conclusion: -

Java AWT (Abstract Window Toolkit) is a set of APIs that provide a platform-independent windowing, graphics, and user interface (UI) toolkit for creating graphical user interfaces in Java programs. Event handling is an important aspect of Java AWT. It is the mechanism that allows the program to respond to user input such as mouse clicks and key presses. The java.awt.event package provides many event classes and listener interfaces for event handling . The steps required to perform event handling include registering the component with the listener, registration methods, and performing event handling code . There are different ways to implement event handling in Java, including implementing ActionListener, using an outer class, or using an anonymous class .