Experiment 10

XII

1. → i) ActionListener
   ii) ItemListener
   iii) KeyListener
   iv) WindowListener

2 → i) keyPressed(KeyEvent ke)
   ii) keyReleased(KeyEvent ke)
   iii) keyTyped(KeyEvent ke)

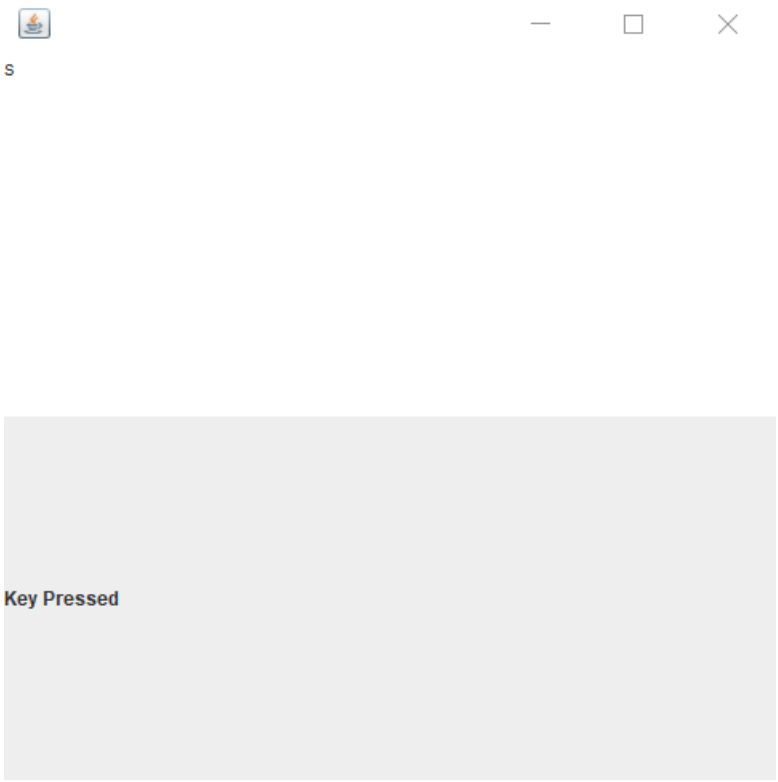
* When a key is typed all 3 methods are generated.

XIII

1. → i) Source object fires an event object when the user check or unchecks a checkbox, if the source was a checkbox.
   ii) The event object contains information about its source object.
   iii) The Listener object is an interface that must be registered as a 'listener' by the source object to be able to respond when the event object was fired, & invoke the handler method of the listener object.

2 → public ~~void~~ abstract void actionPerformed(ActionEvent ae);

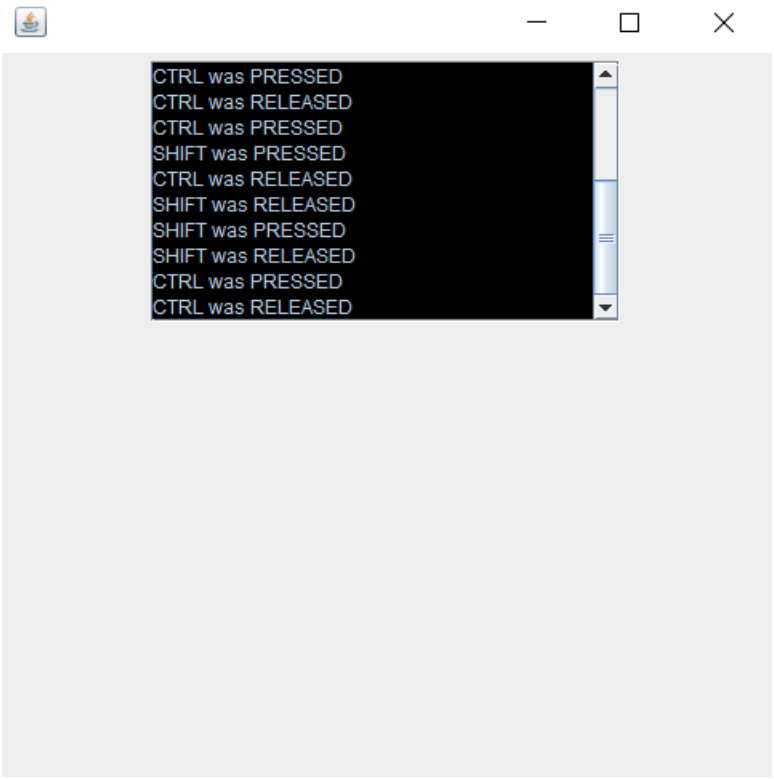
```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class KeyEventDemo0 extends JFrame implements KeyListener {
   JLabel label;

   KeyEventDemo0() {
      Container co = getContentPane();

      label = new JLabel();
      JTextArea ta = new JTextArea();

      co.add(ta);
      co.add(label);

      ta.addKeyListener(this);

      setSize(500, 500);
      setLayout(new GridLayout(2, 1));
      setVisible(true);
   }

   public void keyPressed(KeyEvent e) {
      label.setText("Key Pressed");
   }

   public void keyReleased(KeyEvent e) {
   }

   public void keyTyped(KeyEvent e) {
   }

   public static void main(String[] args) {
      new KeyEventDemo0();
   }
}
```




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

public class KeyEventDemo1 extends JFrame implements KeyListener {
   int vsb = JScrollPane.VERTICAL_SCROLLBAR_ALWAYS;
   int hsb = JScrollPane.HORIZONTAL_SCROLLBAR_NEVER;
   Container co;
   JTextArea ta;

   KeyEventDemo1() {
      co = getContentPane();

      ta = new JTextArea(10, 25);
      ta.setEnabled(false);
      ta.setBackground(Color.black);
      JScrollPane jsp = new JScrollPane(ta, vsb, hsb);

      addKeyListener(this);

      co.add(jsp);

      setSize(500, 500);
      setLayout(new FlowLayout(FlowLayout.CENTER));
      setVisible(true);
   }

   public void keyReleased(KeyEvent ke) {
      String c;
```

```
    int k = ke.getKeyCode();
    switch (k) {
       case KeyEvent.VK_ALT:
          c = "ALT";
          break;
       case KeyEvent.VK_CONTROL:
          c = "CTRL";
          break;
       case KeyEvent.VK_SHIFT:
          c = "SHIFT";
          break;
       default:
          c = "" + ke.getKeyChar();
          break;

    }
    String str = ta.getText() + "\n" + c + " was RELEASED";
    ta.setText(str);
}

public void keyPressed(KeyEvent ke) {
    String c;
    int k = ke.getKeyCode();
    switch (k) {
       case KeyEvent.VK_ALT:
          c = "ALT";
          break;
       case KeyEvent.VK_CONTROL:
          c = "CTRL";
          break;
       case KeyEvent.VK_SHIFT:
          c = "SHIFT";
          break;
       default:
          c = "" + ke.getKeyChar();
          break;

    }
    String str = ta.getText() + "\n" + c + " was PRESSED";
    ta.setText(str);
}

public void keyTyped(KeyEvent ke) {
    String c;
    int k = ke.getKeyCode();
    switch (k) {
       case KeyEvent.VK_ALT:
          c = "ALT";
          break;
       case KeyEvent.VK_CONTROL:
          c = "CTRL";
```

```
            break;
         case KeyEvent.VK_SHIFT:
            c = "SHIFT";
            break;
         default:
            c = "" + ke.getKeyChar();
            break;

      }
      String str = ta.getText() + "\n" + c + " was TYPED";
      ta.setText(str);
   }

   public static void main(String[] args) {
      new KeyEventDemo1();
   }
}
```




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

public class MultiplicationProgram extends JFrame implements ActionListener {
   Container co;
   JTextField jtf1, jtf2, jtf3;
   JButton b1, b2;

   MultiplicationProgram() {
      co = getContentPane();
```

```
        b1 = new JButton("Multiplication");
        jtf1 = new JTextField();
        jtf2 = new JTextField();
        jtf3 = new JTextField();

        co.add(jtf1);
        co.add(jtf2);
        co.add(b1);
        co.add(jtf3);

        b1.addActionListener(this);

        setLayout(new GridLayout(10, 2));

        setTitle("MultiplicationProgram");
        setSize(500, 500);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent ae) {
        float a = Float.parseFloat(jtf1.getText());
        float b = Float.parseFloat(jtf2.getText());
        jtf3.setText("Result : " + Float.toString((a * b)));
    }

    public static void main(String[] args) {
        new MultiplicationProgram();
    }
}
```