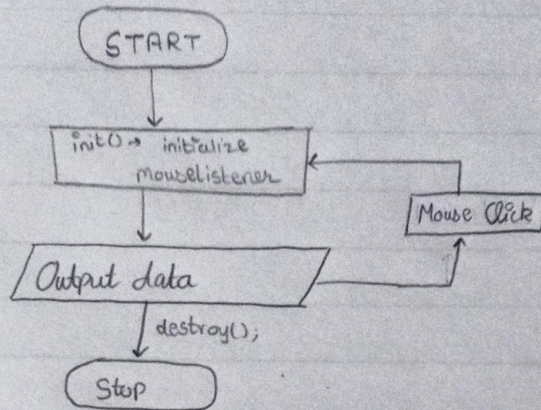


Practical No. 17

Aim: Create, debug and run Java programs based on mouse events and keyboard events.



Aim: Create, debug and run Java programs based on mouse events and keyboard events.

Theory:

What are Events in Java? (Event classes)?

- • All the events in Java have corresponding event classes associated with them.
- Each of these classes is derived from the one single super class, i.e. `EventObject`, it is contained in the `Java.util.` package.
- The event object contains the following two important methods for handling events:
 - 1) `getSource()`;
 - 2) `toString()`;

Java MouseListener Interface:

The java `MouseListener` is notified whenever you change the state of mouse. It is notified against `MouseEvent`. The `MouseListener` is found in `java.awt.event` package. It has five methods:

```
public abstract void mouseClicked(MouseEvent e);
public abstract void mouseEntered(MouseEvent e);
public abstract void mouseExited(MouseEvent e);
public abstract void mousePressed(MouseEvent e);
public abstract void mouseReleased(MouseEvent e);
```


in this class, we have two private methods: `mouseClicked` and `keyPressed`.

They:

- `mouseClicked` is a method in `JPanel` class that is called when the mouse is clicked on the panel.
 - `keyPressed` is a method in `JPanel` class that is called when a key is pressed on the panel.
 - The code snippet below shows how to use these methods.
- ```

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

public class MyPanel extends JPanel {
 // mouseClicked method
 public void mouseClicked(MouseEvent e) {
 System.out.println("Mouse clicked");
 }

 // keyPressed method
 public void keyPressed(KeyEvent e) {
 System.out.println("Key pressed: " + e.getKeyChar());
 }
}

```

The `main` method creates a `JFrame` window and adds the `MyPanel` to it. It also sets the `DefaultCloseOperation` to `HIDE_ON_CLOSE`. The `main` method also calls the `setVisible(true)` method to make the window visible.

Conclusion: In this program, we have used the `MouseListener` and `KeyListener` interfaces to handle mouse and keyboard events.

Hence, I created, debugged and executed java programs based on the concepts of `MouseEvent` and `KeyEvent`.

Code:

Mouse.java

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

/*<applet code="Mouse.class" height="600" width="900"> </applet>*/

public class Mouse extends Applet implements MouseListener{
 int clicks = 0;
 int x = 0;
 int y = 0;
 public void init(){
 addMouseListener(this);
 }

 public void mouseClicked(MouseEvent mb){
 clicks++;
 System.out.println(mb.getClickCount());
 repaint();
 x = mb.getX();
 String str = String.valueOf(x);
 System.out.println(str);
 y = mb.getY();
 str = String.valueOf(y);
 System.out.println(str);
 }

 public void mousePressed(MouseEvent mb){
 showStatus("Mouse is clicked");
 }

 public void mouseReleased(MouseEvent mb){
 showStatus("Mouse click is released");
 }

 public void mouseEntered(MouseEvent mb){

 }

 public void mouseExited(MouseEvent mb){

 }

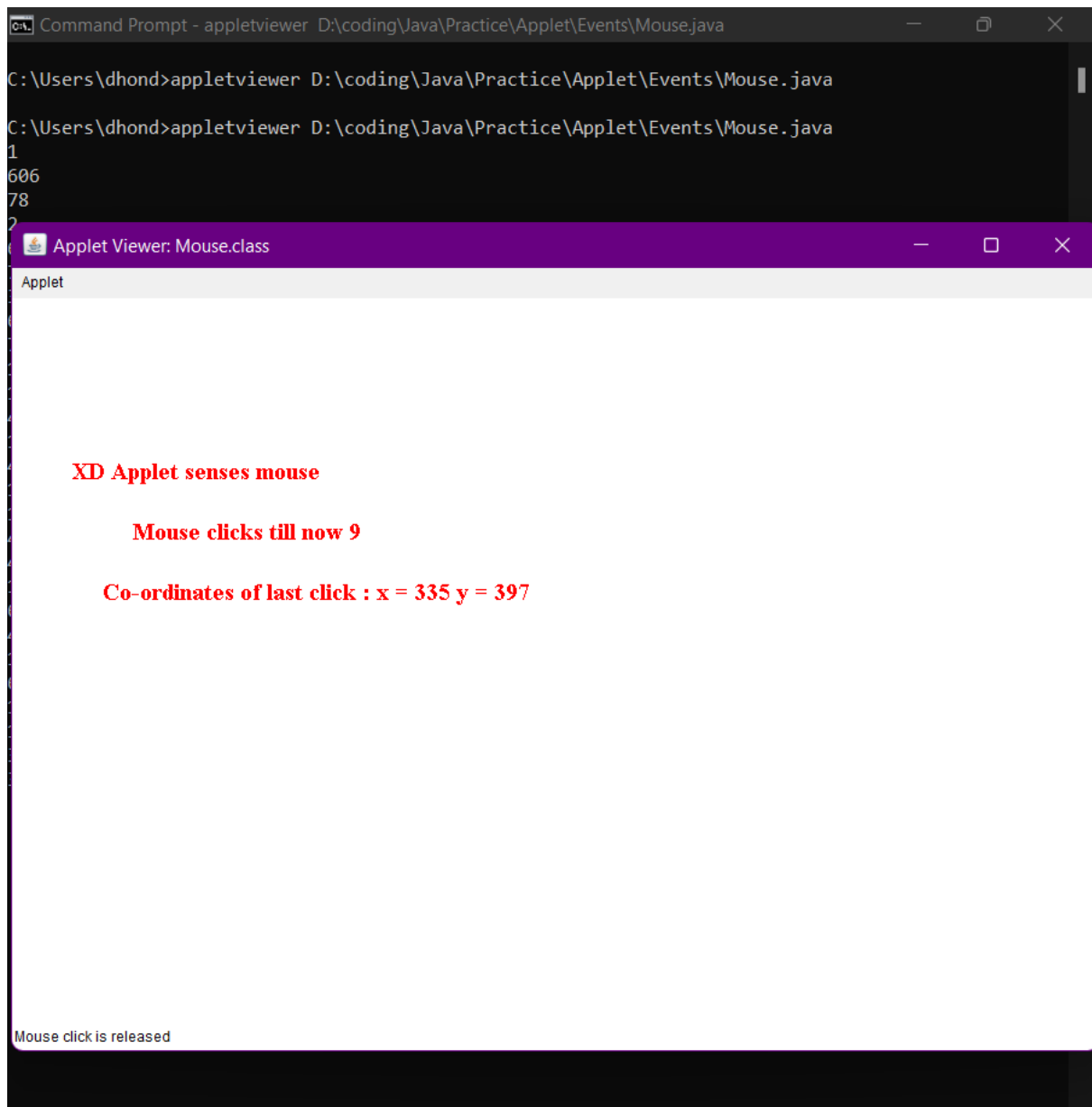
 Font f1 = new Font("Times New Roman",Font.BOLD,20);
```

```
public void paint(Graphics g){
 g.setFont(f1);
 g.setColor(Color.RED);
 g.drawString("Applet senses mouse",50,150);
 g.drawString("Mouse clicks till now " + clicks, 100,200);
 g.drawString("Co-ordinates of last click : x = " + x + " y = " +
y,75,250);
 }

}

// comments
```

Output:



The screenshot shows a web browser window with a Java applet titled "Applet Viewer: Mouse.class". The applet displays the following text in red:

- XD Applet senses mouse**
- Mouse clicks till now 9**
- Co-ordinates of last click : x = 335 y = 397**

At the bottom of the applet, a status bar indicates "Mouse click is released". Above the applet, a Command Prompt window is visible, showing the execution of the command `appletviewer D:\coding\Java\Practice\Applet\Events\Mouse.java` twice, with the output showing the class file being loaded.

Applet using Mouse Event Listener



## Code 2: KeyEventListener

Key.java

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

/*<applet code="Key.class" height="600" width="600"> </applet>*/

public class Key extends Applet implements KeyListener{
 String typed = "";
 public void init(){
 addKeyListener(this);
 }

 public void keyTyped(KeyEvent kb){

 typed += kb.getKeyChar();
 repaint();
 }

 public void keyReleased(KeyEvent kb){
 showStatus("Key on the keyboard is released");
 }

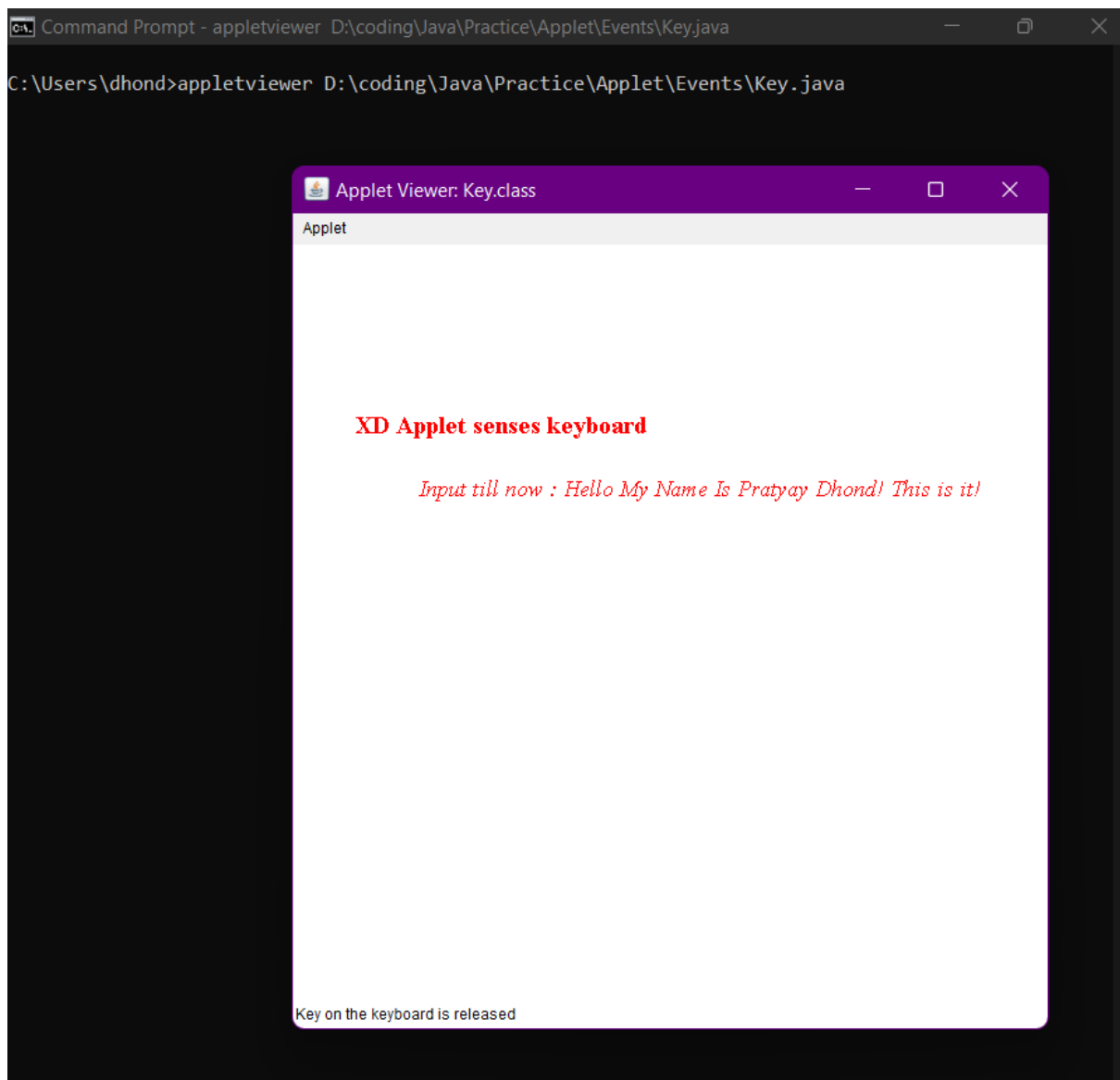
 public void keyPressed(KeyEvent kb){
 showStatus("A key on the keyboard has been pressed");
 }

 Font f1 = new Font("Times New Roman",Font.BOLD,20);
 public void paint(Graphics g){
 g.setFont(f1);
 g.setColor(Color.RED);
 g.drawString("Applet Detects Keyboard",50,150);
 g.setFont(new Font("Serif",Font.ITALIC,18));
 g.drawString("Input till now : " + typed,100,200);
 }

}

// comments
```

Output:



Keyboard event in java output



Conclusion:

Hence, I created, debugged and executed java programs based on the concepts of mouseEvents and keyboardEvents.