JAVA PROGRAMMING LAB (ETCS – 357)

# LAB-9

## Faculty Name : Student Name:

Dr. Sandeep Tayal Udit Agarwal

12914802719

Sem 5

5C7



Maharaja Agrasen Institute of Technology, PSP Area, Sector – 22, Rohini, New Delhi – 110085

# INDEX

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.no** | **Experiments** | **Date of performance** | **Date checked** | **Marks** | | | | | **Total Marks** | **Sign.** |
| **R1** | **R2** | **R3** | **R4** | **R5** |
| 9.1 | WAP that illustrates how to process mouse click, enter, exit, press and release events. The background color changes when the mouse is entered, clicked, pressed, released or exited. | 2.12.21 |  |  |  |  |  |  |  |  |
| 9.2 | WAP that displays your name whenever the mouse is clicked | 2.12.21 |  |  |  |  |  |  |  |  |

## EXPERIMENT 9.1

**AIM : WAP that illustrates how to process mouse click, enter, exit, press and release events. The background color changes when the mouse is entered, clicked, pressed, released or exited.**

## Theory :

**Class :** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical. It represents the set of properties or methods that are common to all objects of one type. A class in java has its methods, variables.

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

#### Methods of MouseListener interface

The signature of 5 methods found in MouseListener interface are given below:

1. public abstract void mouseClicked(MouseEvent e);
2. public abstract void mouseEntered(MouseEvent e);
3. public abstract void mouseExited(MouseEvent e);
4. public abstract void mousePressed(MouseEvent e);
5. public abstract void mouseReleased(MouseEvent e);

**Java Swing :** It is a part of Java Foundation Classes (JFC) that is *used to create window- based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

**Java JFrame :** The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI . Unlike Frame, JFrame has the option to hide or close the window with the help of setDefaultCloseOperation(int) method.

**Source Code :**

import java.awt.event.MouseListener; import java.awt.event.MouseEvent; import java.awt.Color;

import javax.swing.JFrame;

import java.awt.event.ActionListener;

public class Mouse\_Event extends JFrame

{

public static void main(String [] args)

{

JFrame frame=new JFrame(); final int FRAME\_WIDTH=500; final int FRAME\_HEIGHT=600;

frame.setSize(FRAME\_WIDTH,FRAME\_HEIGHT); frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); frame.setVisible(true);

class MouseEnterExitListener implements MouseListener

{

public void mouseEntered(MouseEvent event)

{

System.out.println("ENTER"); frame.getContentPane().setBackground(Color.BLUE);

}

public void mouseExited(MouseEvent event)

{

System.out.println("EXIT"); frame.getContentPane().setBackground(Color.RED);

}

public void mouseReleased(MouseEvent event) { System.out.println("RELEASED"); frame.getContentPane().setBackground(Color.GREEN);

}

public void mouseClicked(MouseEvent event) { System.out.println("CLICKED"); frame.getContentPane().setBackground(Color.BLACK);

}

public void mousePressed(MouseEvent event) { System.out.println("PRESSED"); frame.getContentPane().setBackground(Color.YELLOW);

}

}

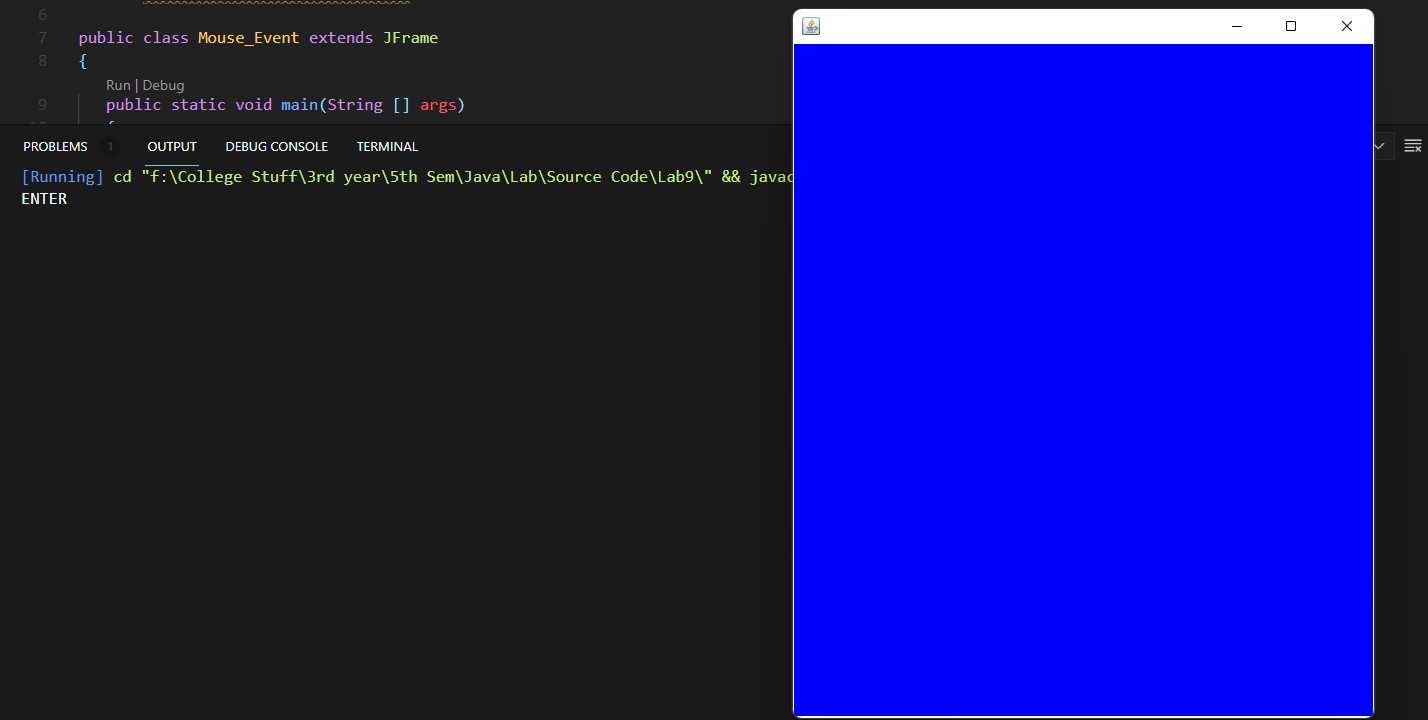
MouseListener listener = new MouseEnterExitListener(); frame.addMouseListener(listener);

}

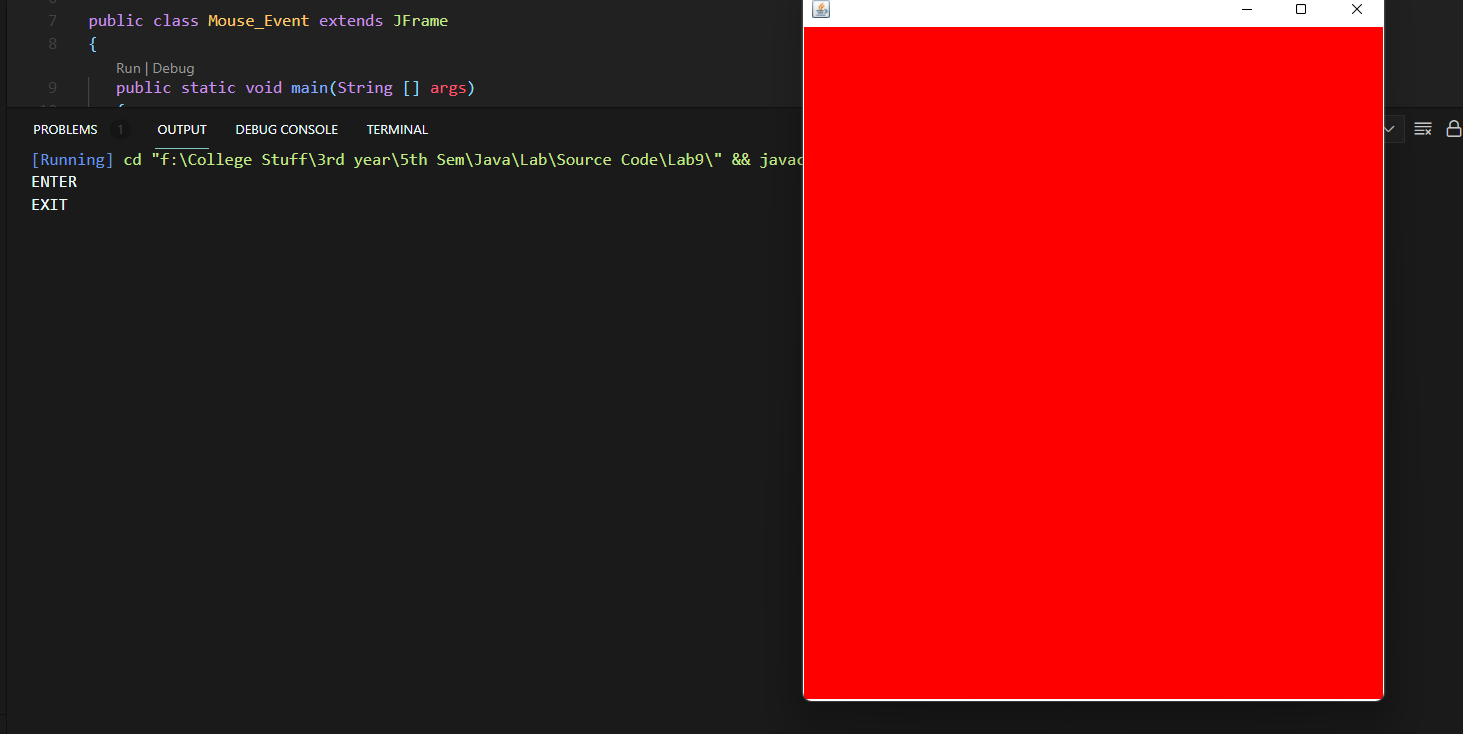
}

### Output :

**On Mouse Entering**



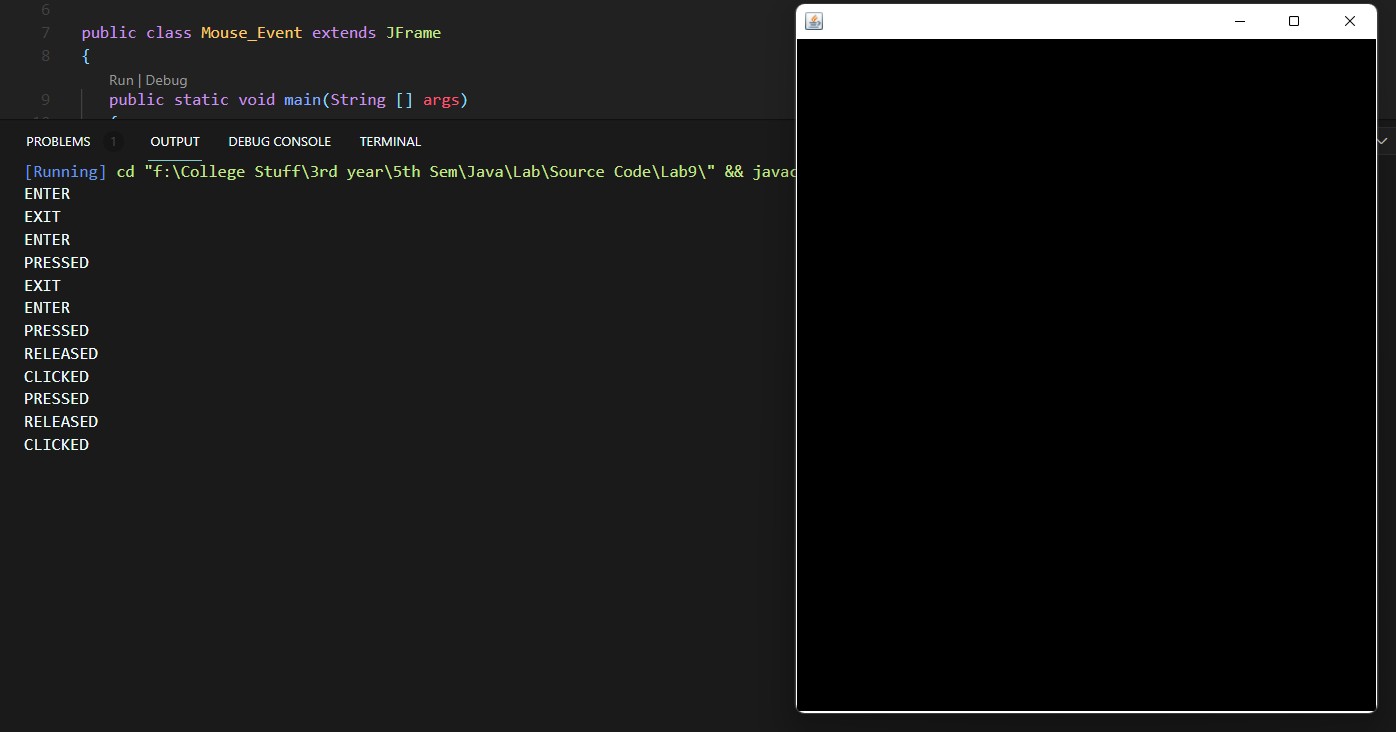
**On Mouse Exiting**



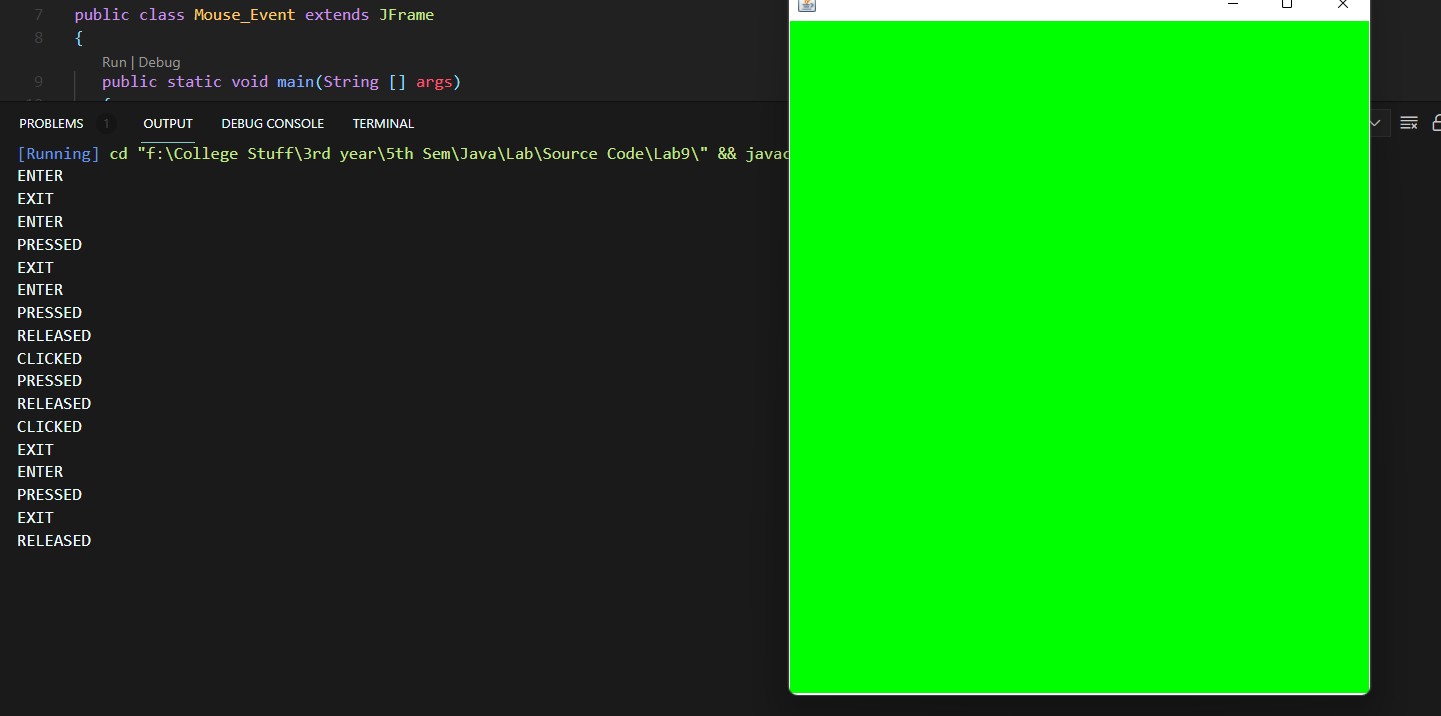
**On Mouse Press**



**On Mouse Click**



**On Mouse Release**



## EXPERIMENT 9.2

**AIM : WAP that displays your name whenever the mouse is clicked**

## Theory :

**Class :** A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical. It represents the set of properties or methods that are common to all objects of one type. A class in java has its methods, variables.

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

#### Methods of MouseListener interface

The signature of 5 methods found in MouseListener interface are given below:

1. public abstract void mouseClicked(MouseEvent e);
2. public abstract void mouseEntered(MouseEvent e);
3. public abstract void mouseExited(MouseEvent e);
4. public abstract void mousePressed(MouseEvent e);
5. public abstract void mouseReleased(MouseEvent e);

**Java Swing :** It is a part of Java Foundation Classes (JFC) that is *used to create window- based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

**Java JFrame :** The javax.swing.JFrame class is a type of container which inherits the java.awt.Frame class. JFrame works like the main window where components like labels, buttons, textfields are added to create a GUI . Unlike Frame, JFrame has the option to hide or close the window with the help of setDefaultCloseOperation(int) method.

**Source Code :**

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

public class Mouse\_click implements MouseListener, ActionListener {

static JFrame frame; static JTextField text;

public static void main(String[] args) {

frame = new JFrame("Display Name on mouse click"); frame.setBackground(Color.white); frame.setSize(500, 500);

frame.setLayout(null); text = new JTextField();

text.setBounds(0, 0, 500, 50); frame.add(text);

JButton exit = new JButton("Exit"); exit.setBounds(220, 235, 60, 30); frame.add(exit);

Mouse\_click obj = new Mouse\_click(); frame.addMouseListener(obj); exit.addActionListener(obj); frame.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) { frame.dispose();

}

@Override

public void mouseEntered(MouseEvent e) { text.setText("");

}

@Override

public void mouseExited(MouseEvent e) { text.setText("");

}

@Override

public void mouseReleased(MouseEvent e) { text.setText("");

}

@Override

public void mousePressed(MouseEvent e) { text.setText("");

}

@Override

public void mouseClicked(MouseEvent e) { text.setText("");

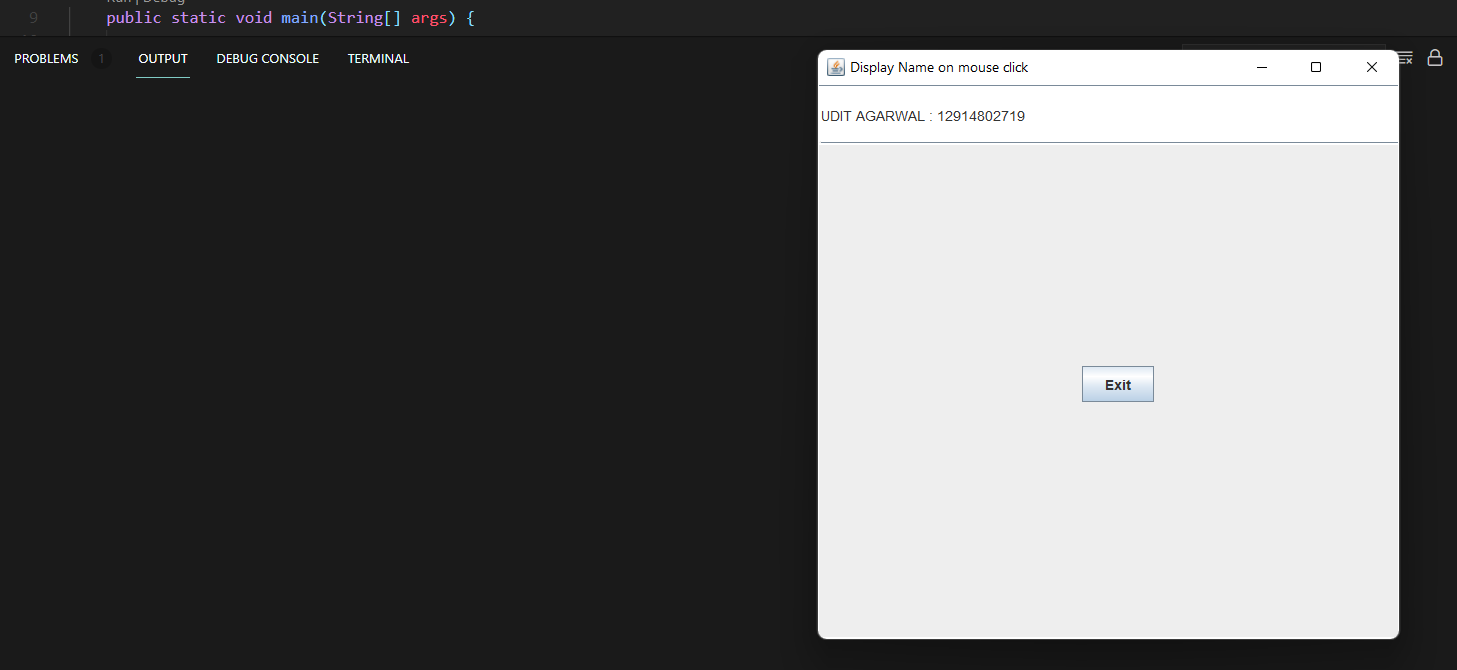
text.setText("UDIT AGARWAL : 12914802719");

}

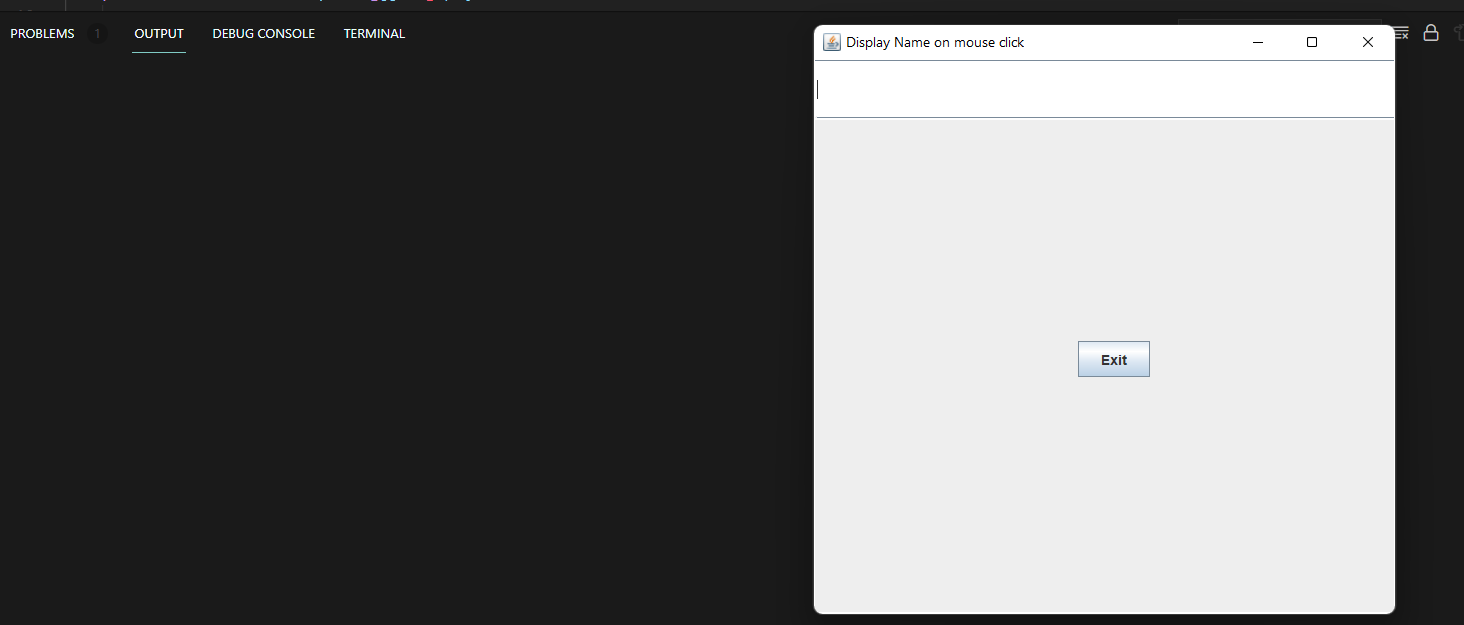
}

### Output :

**On Mouse Click :**



**Without Mouse Click :**



## VIVA QUESTIONS :

**Q1) Difference between Swing and Awt?**

**Ans** AWT are heavy-weight componenets. Swings are light-weight components. Hence swing works faster than AWT.

**Q2). What Are The Different Types Of Controls In Awt? Ans.** The AWT supports the following types of controls:

Labels, Pushbuttons, Checkboxes, Choice lists, Lists, Scroll bars, Text componentsThese controls are subclasses of component.

**Q3) What Are The Benefits Of Swing Over Awt? Ans.**

* Swing components are light weight.
* We can have a pluggable look and feel feature which shows us how they appear in other platforms.
* We can add images to Swing components. We have toolbars and tooltips in Swing.