|  |  |
| --- | --- |
| **Name – Prerna Sunil Jadhav** | **SAP ID - 60004220127** |

**Experiment No – 15**

**AIM: Develop simple swing applications and complex GUI using Java Swing classes. (CO6)**

**THEORY:**

Swing in Java is a lightweight GUI toolkit which has a wide variety of widgets for building optimized window based applications. It is a part of the JFC( Java Foundation Classes). It is build on top of the AWT API and entirely written in java. It is platform independent unlike AWT and has lightweight components. It becomes easier to build applications since we already have GUI components like button, checkbox etc. This is helpful because we do not have to start from the scratch. Different methods of JFrame class are : public void add(Component c) : Inserts a component on this component. public void setSize(int width,int height) : Sets the size (width and height) of the component. public void setLayout(LayoutManager m) : Defines the layout manager for the component. public void setVisible(boolean status) : Changes the visibility of the component, by default false. An object of the java.awt.Font class represents a font in a Java program

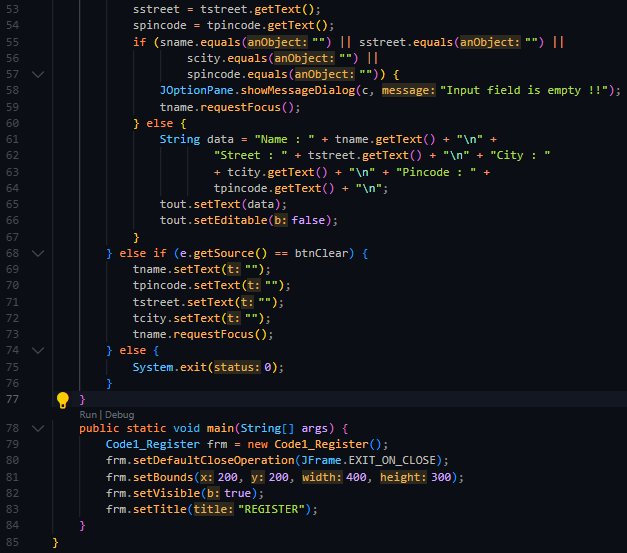
**Code (i): Write a program to create a window with four text fields for the name, street, city and pin code with suitable labels. Also windows contains a button MyInfo. When the user types the name, his street, city and pincode and then clicks the button, the types details must appear in Arial Font with Size 32, Italics.**

**Text

Description automatically generated**

**Text

Description automatically generated**

****

**Text

Description automatically generated**

**OUTPUT:**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**Theory:**

Interface ActionListener : The listener interface for receiving action events. The class that is interested in processing an action event implements this interface, and the object created with that class is registered with a component, using the component's addActionListener method. When the action event occurs, that object's actionPerformed method is invoked. The EventObject contains the getSource( ) method.Suppose you have many buttons in your application. So, you can find which button is clicked by using the getSource() method. The getSource( ) method returns the source of the event. Java CollationElementIterator setText(String source) Set a new string over which to iterate.

**Code (ii): WA applet with 4 swing buttons with suitable texts on them. When the user presses a button a message should appear in the label as to which button was pressed by the user**

**Text

Description automatically generated**

**Text

Description automatically generated**

**OUTPUT:**

Graphical user interface

Description automatically generated Graphical user interface

Description automatically generated