Assignment 14

Name: Priyansh Salian,

Roll No: 2003148

Program Statement:

1. **Program to create a GUI based Application**

   Create a registration form containing required fields .

   The form should have all the studied components

   The form should have minimum two buttons “Submit” and “Cancel”….giving appropriate messages at corresponding click Theory:

GUI **(Graphical User Interface)** in Java is an **easy-to-use visual experience builder** for Java applications. It is mainly made of graphical components like buttons, labels, windows, etc. through which the user can interact with an application. GUI plays an important role to build easy interfaces for Java applications.

Java AWT (Abstract Window Toolkit) is an API to develop Graphical User Interface (GUI) or windows-based applications in Java.

Java AWT components are platform-dependent i.e. components are displayed according to the view of operating system. AWT is heavy weight i.e. its components are using the resources of underlying operating system (OS).

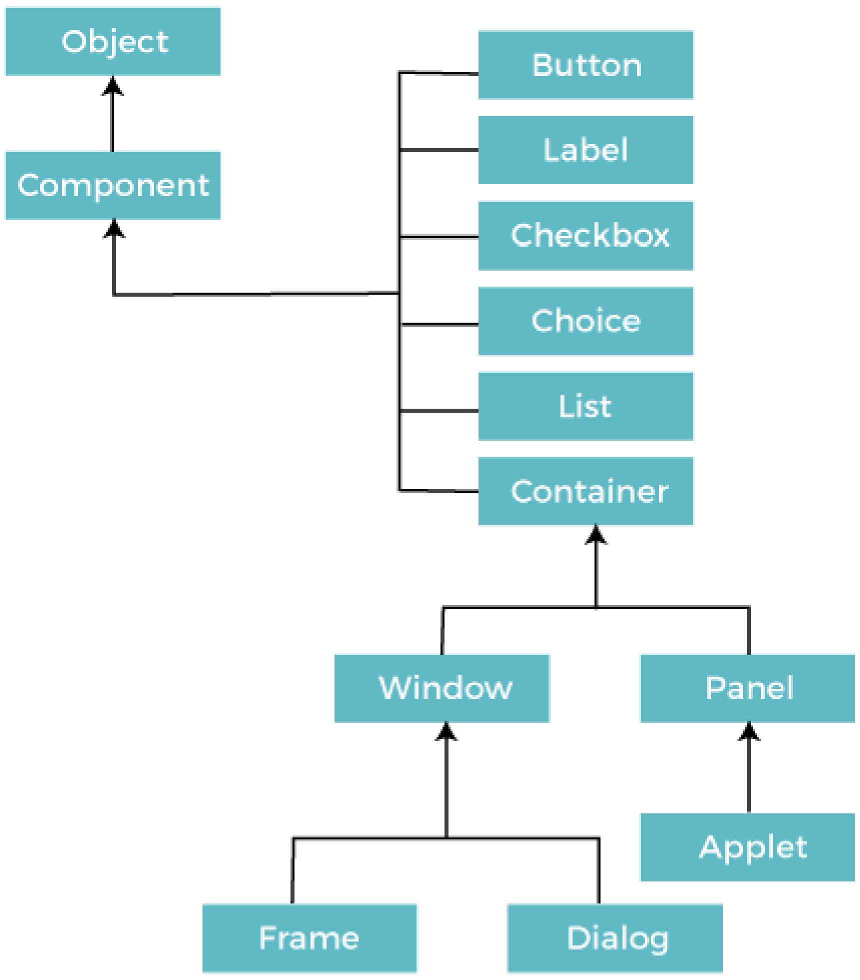
The java.awt package provides classes for AWT API such as TextField , Label, TextArea , RadioButton, CheckBox , Choice , List, etc.

Java AWT calls the native platform calls the native platform (operating systems) subroutine for creating API components like TextField, ChechBox, button, etc.

For example, an AWT GUI with components like TextField, label and button will have different look and feel for the different platforms like Windows, MAC OS, and Unix. The reason for this is the platforms have different view for their native components and AWT directly calls the native subroutine that creates those components.

In simple words, an AWT application will look like a windows application in Windows OS whereas it will look like a Mac application in the MAC OS.

The hierarchy of Java AWT classes are given below.



All the elements like the button, text fields, scroll bars, etc. are called components. In Java AWT, there are classes for each component as shown in above diagram. In order to place every component in a particular position on a screen, we need to add them to a container.

The Container is a component in AWT that can contain another components like buttons

, textfields, labels etc. The classes that extends Container class are known as container such as Frame, Dialog and Panel.

It is basically a screen where the where the components are placed at their specific locations. Thus it contains and controls the layout of components.

Types of containers:

There are four types of containers in Java AWT:

1. Window
2. Panel
3. Frame
4. Dialog

Window

The window is the container that have no borders and menu bars. You must use frame, dialog or another window for creating a window. We need to create an instance of Window class to create this container.

Panel

The Panel is the container that doesn't contain title bar, border or menu bar. It is generic container for holding the components. It can have other components like button, text field etc. An instance of Panel class creates a container, in which we can add components.

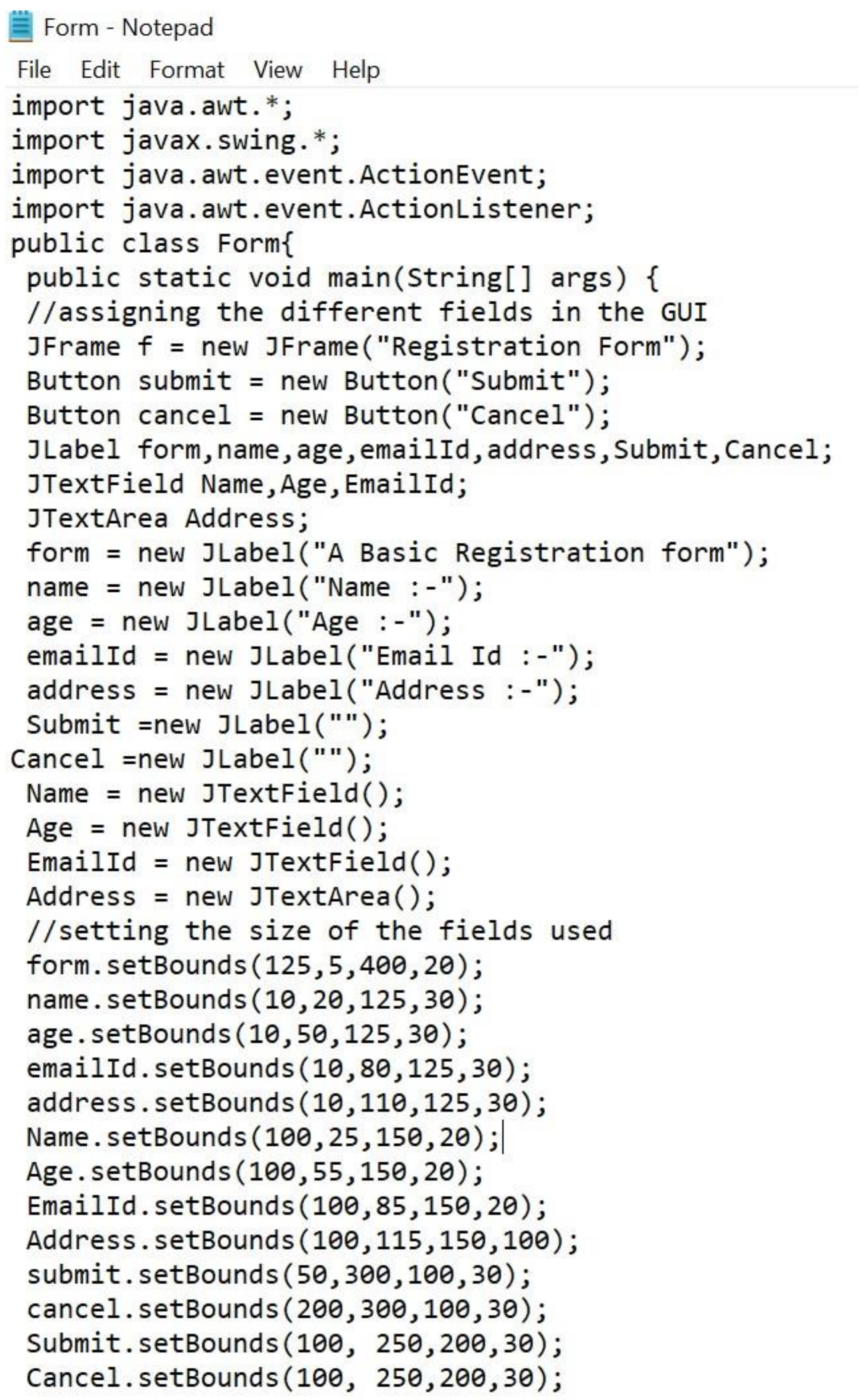
Frame

The Frame is the container that contain title bar and border and can have menu bars. It can have other components like button, text field, scrollbar etc. Frame is most widely used container while developing an AWT application.

To create simple AWT example, you need a frame. There are two ways to create a GUI using Frame in AWT.

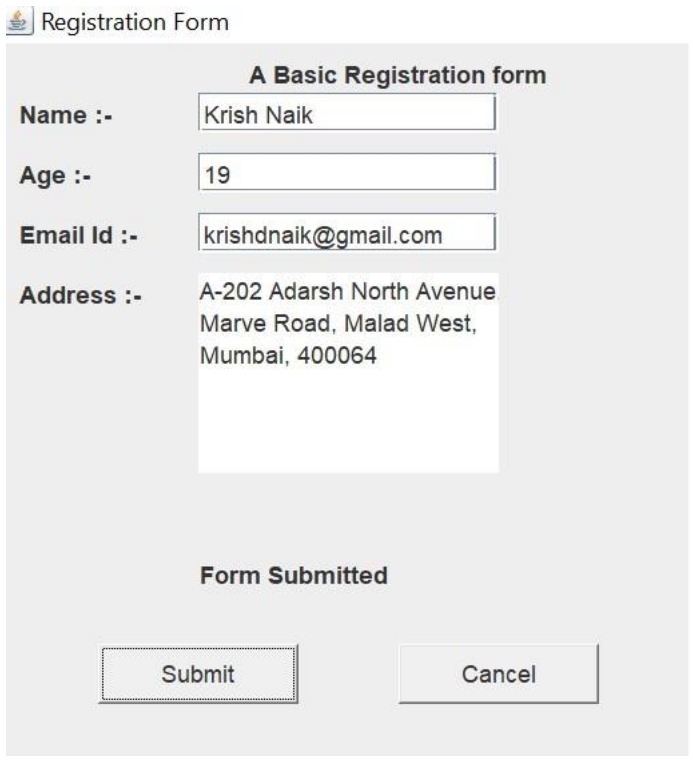
1. By extending Frame class (inheritance)
2. By creating the object of Frame class (association)

Program:



```
import java.awt.*;
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class Form{
 public static void main(String[] args) {
 //assigning the different fields in the GUI
 JFrame f = new JFrame("Registration Form");
 Button submit = new Button("Submit");
 Button cancel = new Button("Cancel");
 JLabel form,name,age,emailId,address,Submit,Cancel;
 JTextField Name,Age,EmailId;
 JTextArea Address;
 form = new JLabel("A Basic Registration form");
 name = new JLabel("Name :-");
 age = new JLabel("Age :-");
 emailId = new JLabel("Email Id :-");
 address = new JLabel("Address :-");
 Submit =new JLabel("");
Cancel =new JLabel("");
 Name = new JTextField();
 Age = new JTextField();
 EmailId = new JTextField();
 Address = new JTextArea();
 //setting the size of the fields used
 form.setBounds(125,5,400,20);
 name.setBounds(10,20,125,30);
 age.setBounds(10,50,125,30);
 emailId.setBounds(10,80,125,30);
 address.setBounds(10,110,125,30);
 Name.setBounds(100,25,150,20);
 Age.setBounds(100,55,150,20);
 EmailId.setBounds(100,85,150,20);
 Address.setBounds(100,115,150,100);
 submit.setBounds(50,300,100,30);
 cancel.setBounds(200,300,100,30);
 Submit.setBounds(100, 250,200,30);
 Cancel.setBounds(100, 250,200,30);
 //Adding elements in the screen
 f.add(form);
f.add(submit);
 f.add(name);
 f.add(Name);
 f.add(age);
 f.add(Age);
 f.add(emailId);
 f.add(EmailId);
 f.add(address);
 f.add(Address);
 f.add(cancel);
 f.add(Submit);
 f.add(Cancel);
 //Size of the GUI,layout,visibility
 f.setSize(500,500);
 f.setLayout(null);
 f.setVisible(true);
 submit.addActionListener(new ActionListener()
 {
 @Override
 public void actionPerformed(ActionEvent e){
 Submit.setVisible(true);
 Submit.setText("Form Submitted");
 Cancel.setVisible(false);
}
 });
 cancel.addActionListener(new ActionListener()
 {
 @Override
 public void actionPerformed(ActionEvent e){
 Cancel.setVisible(true);
 Cancel.setText("Form Cancelled");
 Submit.setVisible(false);
 }
 });
 }
}
```

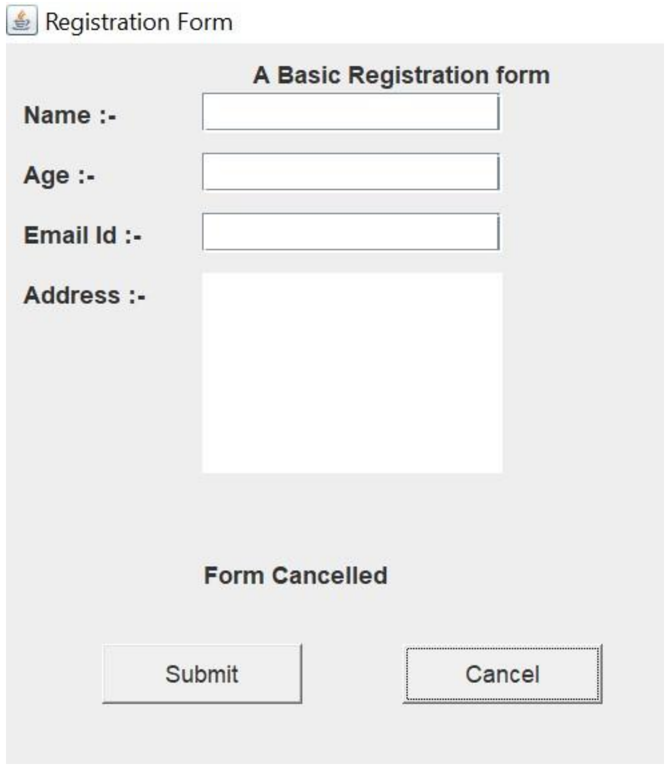
Output:


Registration Form

A Basic Registration form

Name :-	Krish Naik
Age :-	19
Email Id :-	krishdnaik@gmail.com
Address :-	A-202 Adarsh North Avenue Marve Road, Malad West, Mumbai, 400064

Form Submitted


Registration Form

A Basic Registration form

Name :-	
Age :-	
Email Id :-	
Address :-	

Form Cancelled