# Assignment - IV

Title :- Transform the system from command line system to GUI based application

Problem Statement :- GUI programming

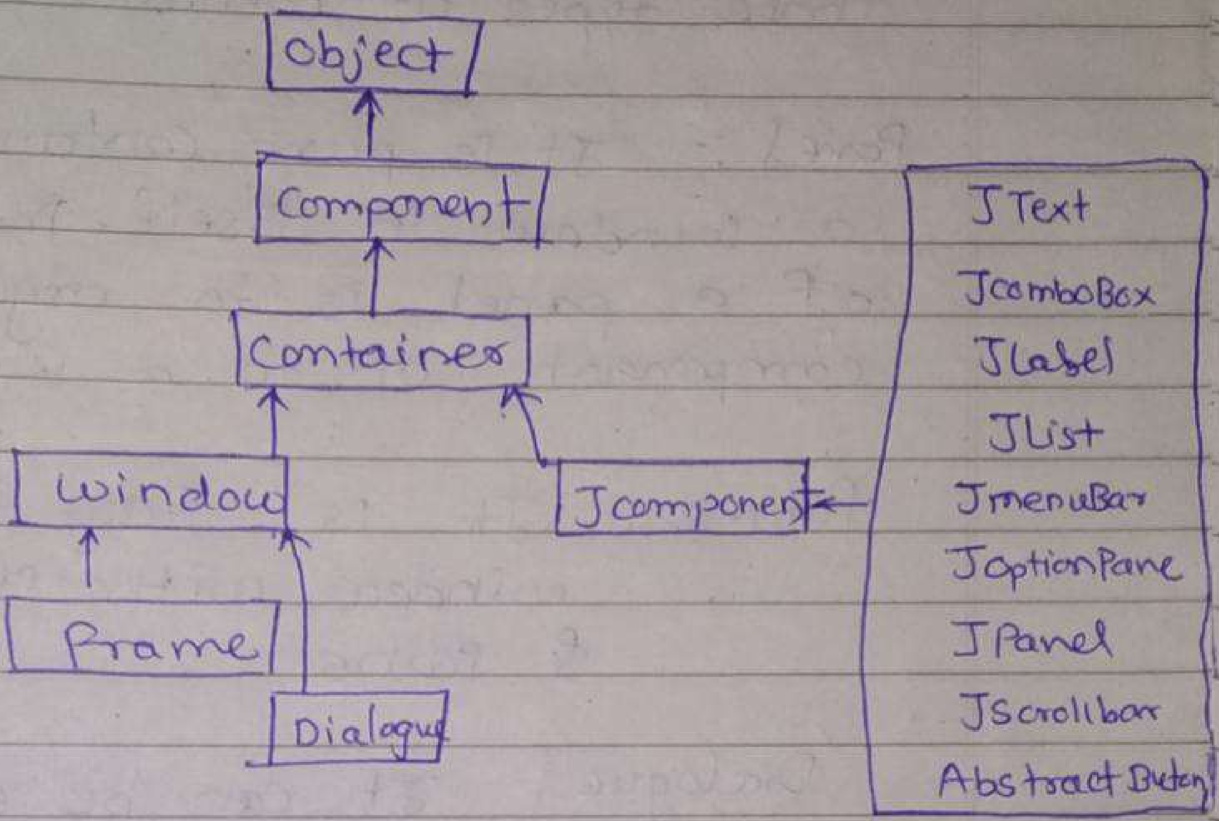
Objective :- Understand the implementation of swing class.

Outcome :- After completion of this assignment students can evaluate & analyze the problem & understand the GUI concepts in java,

S/W & H/W requirement :-
64 bit Fedora OS, Java developement kit, Eclipce IDE,

Concept Related Theory :-
Java swing provides platform independent & lightweight components The java-swing package provides classes for java Swing API such as JButton, JTextfield, JTextArea, Jlabel, JPanel, JradioButton, etc.

# Java Swing class Hierarchy



All components in swing are Jcomponent which can be added to container classes.

## Building GUI application.

i) Identify user requirement for GUI.
ii) Map the requirements to swing approach
iii) Group UI ~~ente~~ components.
iv) Define UI structure
v) Select suitable event to handle.
vi) Link events to execution ~~state~~ logic.

Container Class :

container classes are classes that can have other components on it.

we need atleast one container object three types of containers.

Panel :- It is pure container & is not a window in itself. The sole purpos of a panel is to organize the component on to a window.

Frame :- It is fully functioning window with its own title & icons.

Dialogue :- It can be ~~so~~ thought of as a popup window that pops out when message has to be displayed. It is not ~~&~~ a fully functioning window like the frame.

JButton :- The Jbutton class is used to create labeled button that has platform independent implementation. The application result in same action when the Button is pushed. It inherits AbstractButton class.

Jlabel : The object of Jlabel class is a component for placing text in a container, It is used to display a single line of read only text. The Text can be changed by an application but a user cannot edit it directly. It inherits Jcomponents class.

J TextArea :- The object of J TextArea class is a multiline region that displays text. It allows the editing of multiple line text. It inherits JText components class,

J checkbox :- It is used to turn o~~ff~~ an option on or off. clicking an check-box changes its state.

JRadio Button :- It is used to choose one option from multiple options. It should be added in ButtonGroup to select one radio button only.

JcomboBox :- A component that combines a button or editable field & a dropdown list. The User can select value from the dropdown list.

Event Listner:-

When button is clicked, it creates an action event object & invokes action performed (ActionEvent) method of ActionListner interface. Event source needs a reference to the object of Event handler so that it can call it's method.

Algorithm :-

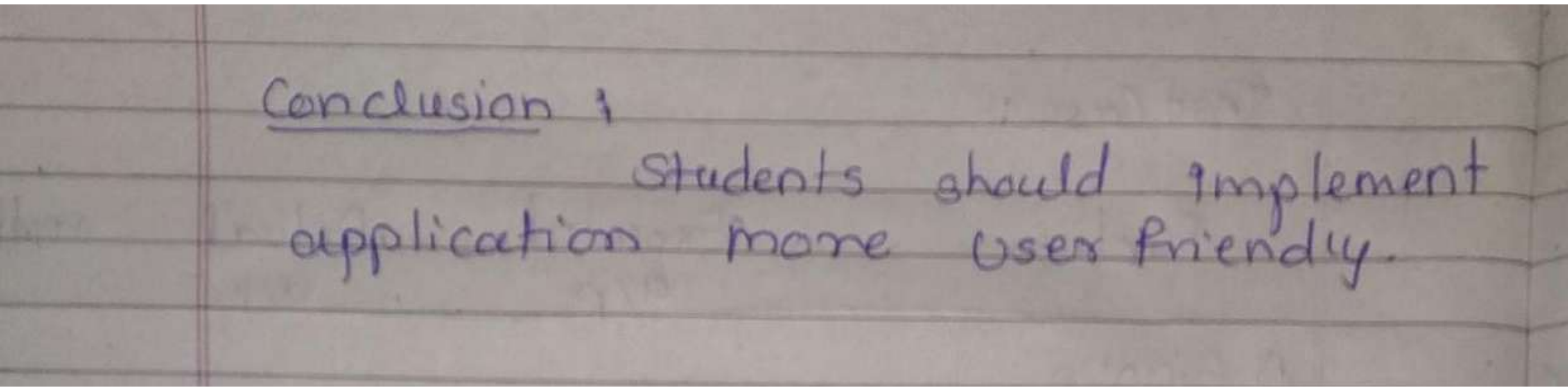
```
public  Ask () {
    // This constructor invoke the
    // welcome screen & asks for
    // either login or signUp.
    // It will  have login & signUp
    // button which invokes respective
    // frames.
}
public login () {
// This constructor invoke the login
   form frame which take input
   credentials.
// If credentials match as per
   database then invokes new
     main screen after submit
}
```

```
public SignUp () {
  // This constructor invokes window to take
    input form for signUp & submit
     button.
  // This will add new employee data
   to database
}
public Main () {
    // This one invokes userframe
    & displays features for particular
      user.
   // Based on eventlistners next frame
      is generated.
 }
 public Admin () {
    // It invokes admin frame &
  shows features for admin user.
   // According to eventlistners next
     frames are generated.
 }
 public mainPrj () {
     // It invokes a user frame showing
   All projects assigned to user &
   its details
   // According eventlistners next
     frame is gnemerated.
 }
```

```
public  PrjCreate() {
    // Invokes frame having input
       form to take project details
       & assigned to a employee.
    // changes are made in database
       Accordingly.
}
public  prjDelete() {
    // Invokes frame for Admin to
       delete a existing project.
    // Input is taken & Changes to
       database are done accordingly.
}
public  deadlineUpdate(){
    // Another frame for admin is
  invoked to change the deadlines for
  project.
  // According to input changes made
     to database
  }
public  EmpDelete() {
 // invokes a frame for admin to
 remove an employee from portal
   // Input is taken & changes
 made to database accordingly.
}
```

Test Case :-

| function | Expected o/p | Actual o/p | result |
|---|---|---|---|
| 1) SignUp<br>User.<br>Name: "Sunveg"<br>Designat$^{n}$: "soft. Engg."<br>Experience : 2 (years)<br>mail : "sunveg@gmail.com" | Message:<br>"SignUp Successful.<br>Password is 'sunveg@123'! | same | Success. |
| 2) Create<br>Project.<br>ProjectName:<br>'Web App'<br>Project Detail:<br>'Develope an web app'<br>Deadline:<br>'20201231'<br>Employee:<br>'Sunveg'. | Message:<br>"Web App project Assigned to User Suneg!" | Same | Succees. |



Conclusion :

Students should implement application more user friendly.