Object Orientated Programming

Programming Day

Instructor:

Mr. Samyan Qayyum Wahla

Learning Objectives:

- To get familiar Event driven Programming and GUI components

CLO:

- CLO1

Registration Number:

____

Name:

____

Guidelines/Instructions:

- Use of NetBeans is must in this lab.
- Write well commented code.
- Name of variables should be meaningful.
- Use camel case for the naming convention
- Code should be well formatted
- Create meaningful variable names. Add comments for readability. Indent each line of your code.
- Plagiarism/Cheating is highly discouraged by penalizing to both who tried and one who shared his/her code.

Reading Content:

- Lecture content of the week 3.

Event Driven Programming:

Event-driven programming is a programming paradigm in which the flow of program execution is determined by events - for example a user action such as a mouse click, key press, or a message from the operating system or another program. An event-driven application is designed to detect events as they occur, and then deal with them using an appropriate event-handling procedure. The idea is an extension of interrupt-driven programming of the kind found in early command-line environments such as DOS, and in embedded systems (where the application is implemented as firmware).

Event-driven programs can be written in any programming language, although some languages(Visual Basic for example) are specifically designed to facilitate event-driven programming, and provide an integrated development environment (IDE) that partially automates the production of code, and provides a comprehensive selection of built-in objects and controls, each of which can respond to a range of events. Virtually all object-oriented and visual languages support event-driven programming. Visual Basic, Visual C++, Java and C# are examples of such languages.

A visual programming IDE such as C# provides much of the code for detecting events automatically when a new application is created. The programmer can therefore concentrate on issues such as interface design, which involves adding controls such as command buttons, text boxes, and labels to standard forms (a form represents an application's workspace or window). Once the user interface is substantially complete, the programmer can add event-handling code to each control as required.

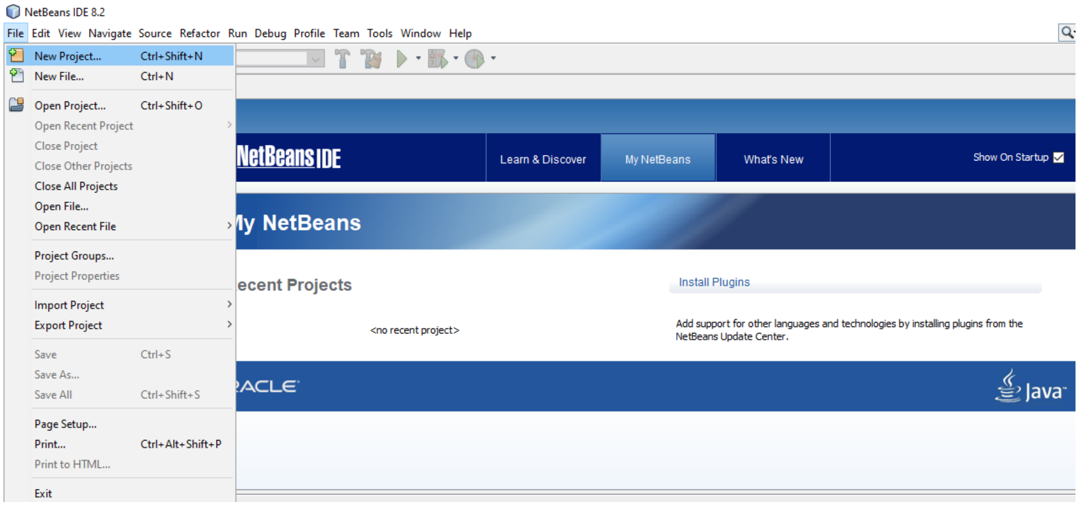
Events:

An event can be mouse click, key press, or a message from the operating system or another program.

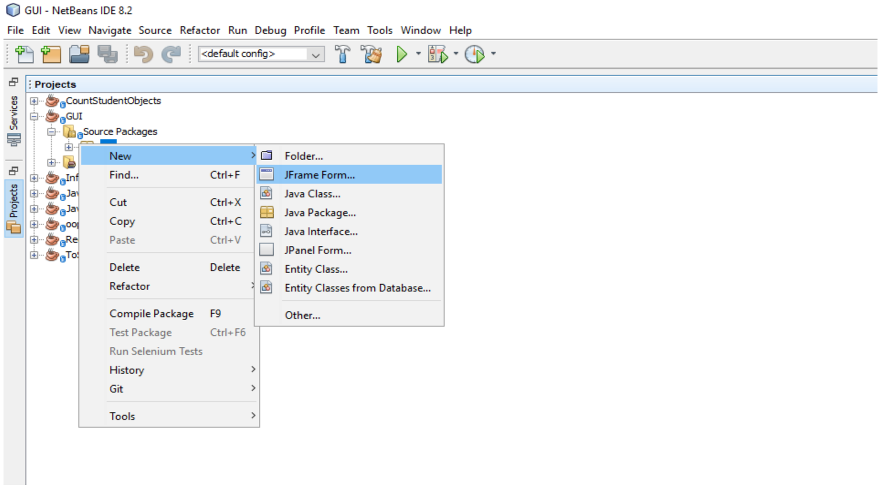
Frameworks

1. **A program that shows Hello world in popup frame at button click**

Step 1:
Open net beans and click on new project.



Step 2:
Give any project name to your project and then click on finish button. Once the project is created, right click on project name and click New then click on JFrame form. JFRame Form contains the following component and their properties.



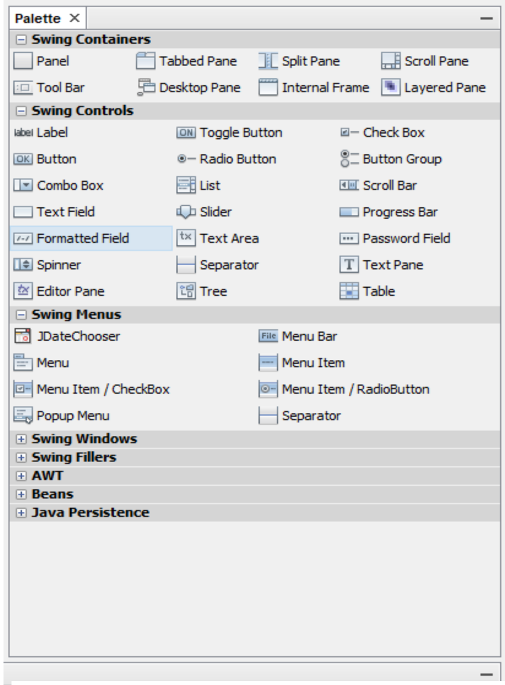
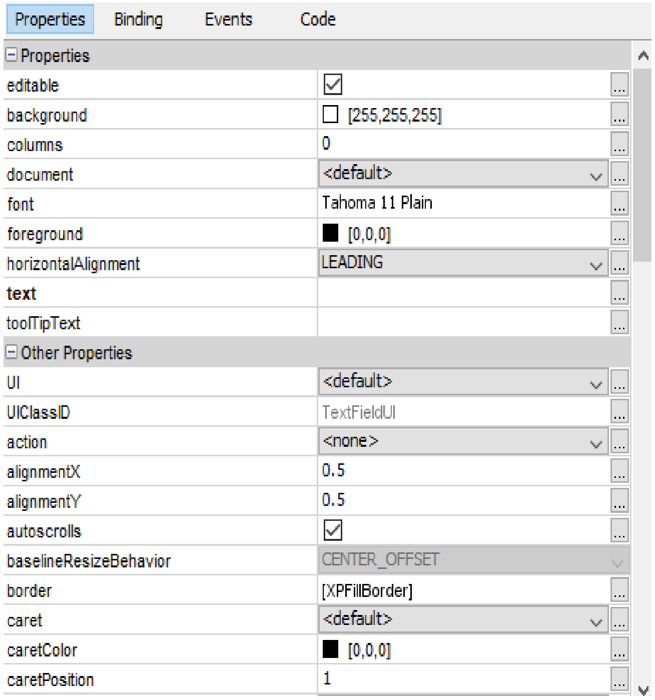


Figure 2: GUI Component

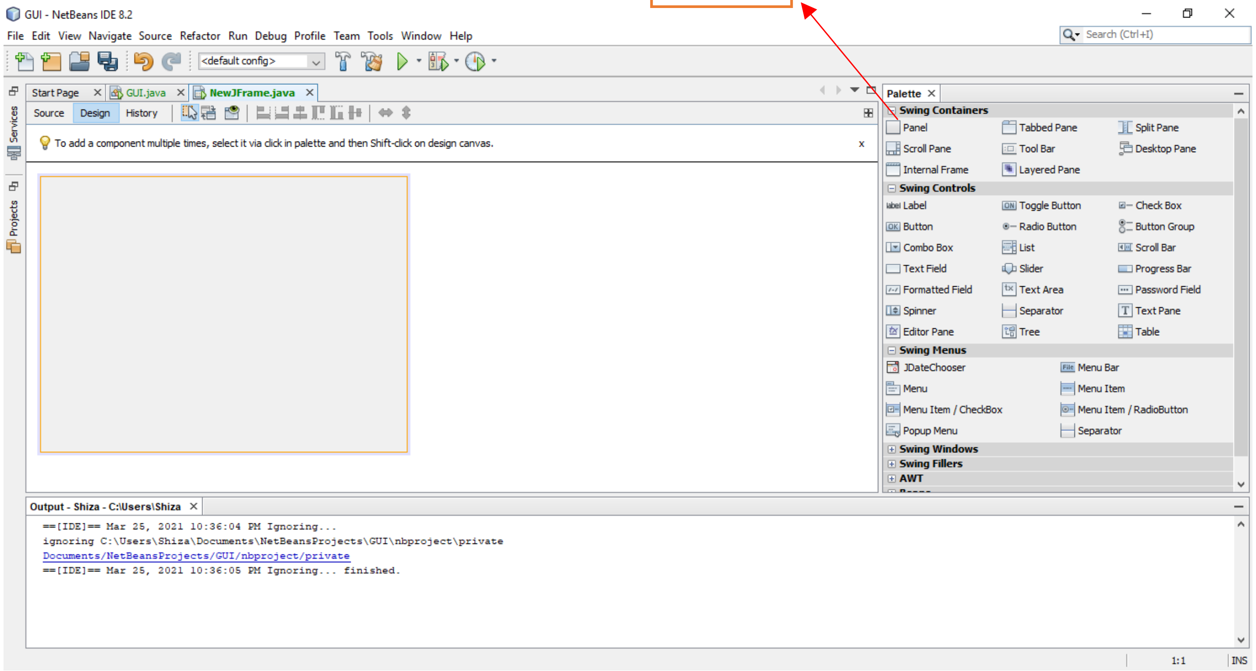


Properties	
editable	☑
background	□ [255,255,255]
columns	0
document	<default>
font	Tahoma 11 Plain
foreground	■ [0,0,0]
horizontalAlignment	LEADING
text	
toolTipText	
Other Properties	
UI	<default>
UIClassID	TextFieldUI
action	<none>
alignmentX	0.5
alignmentY	0.5
autoscrolls	☑
baselineResizeBehavior	CENTER_OFFSET
border	[XPFillBorder]
caret	<default>
caretColor	■ [0,0,0]
caretPosition	1

Figure 1: GUI Control and their properties

Step 3:

Frame creation



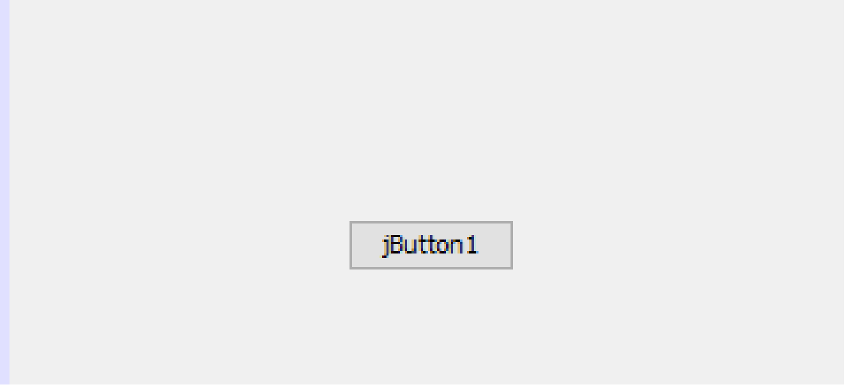
To add a component multiple times, select it via click in palette and then Shift-click on design canvas.

```
==[IDE]== Mar 25, 2021 10:36:04 PM Ignoring...
ignoring C:\Users\Shiza\Documents\NetBeansProjects\GUI\nbproject\private
Documents/NetBeansProjects/GUI/nbproject/private
==[IDE]== Mar 25, 2021 10:36:05 PM Ignoring... finished.
```

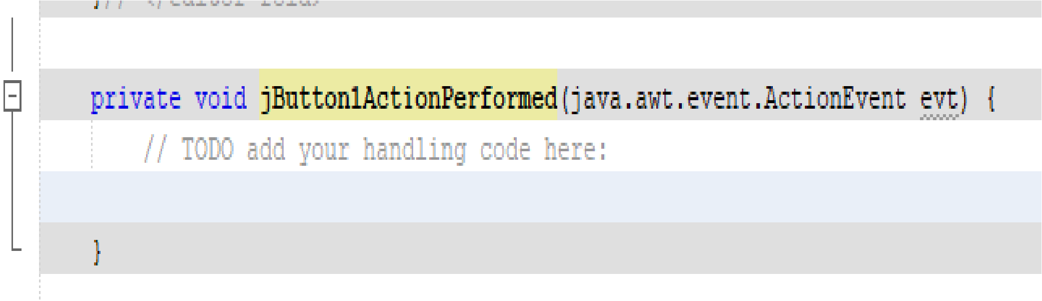
Palette shows in the right side. Inside palette, there are many components. Drag and drop the component button from the palette into frame.

Step 4:

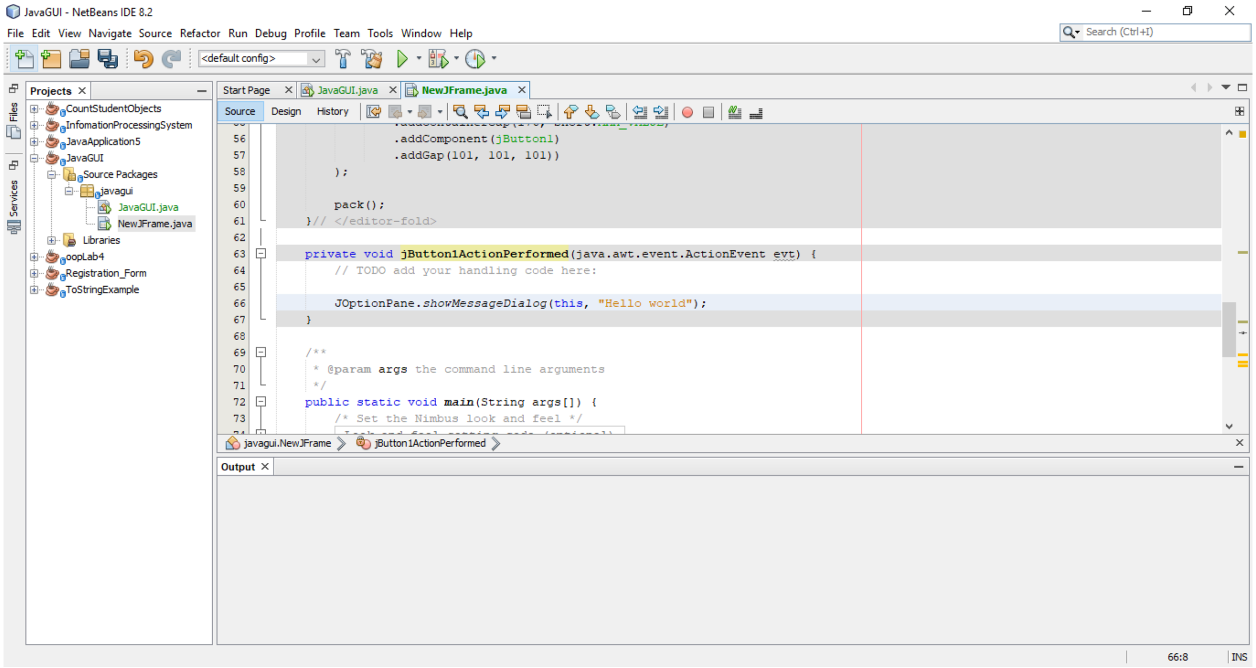
Add button on frame.



Step 5:
Double click on button for action perform event of that button. You will see the jButton1ActionPerformed function. (This function will execute when jButton1ActionPerformed event will occur).



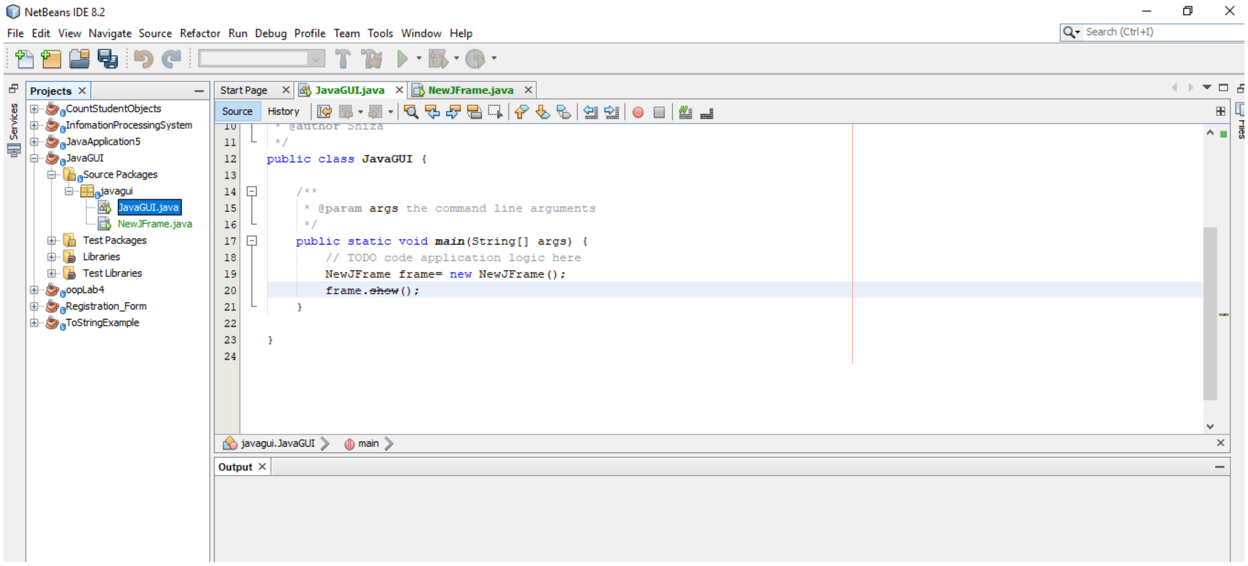
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

}
```



```
                .addComponent(jButton1)
                .addGap(101, 101, 101))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:

        JOptionPane.showMessageDialog(this, "Hello world");
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
```

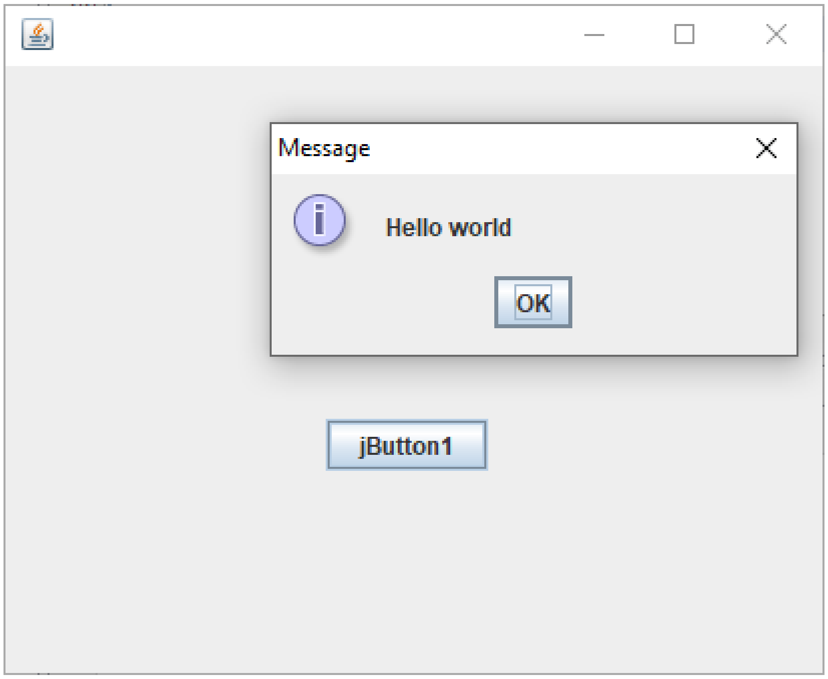
Step 6: Before run project, call frame in main for display frame directly when run project.

Output:



```
 * @author Shiza
 */
public class JavaGUI {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        NewJFrame frame= new NewJFrame();
        frame.show();
    }

}
```

1. A program that shows Hello world + Name in popup frame at button click



Hello world

Step 1:

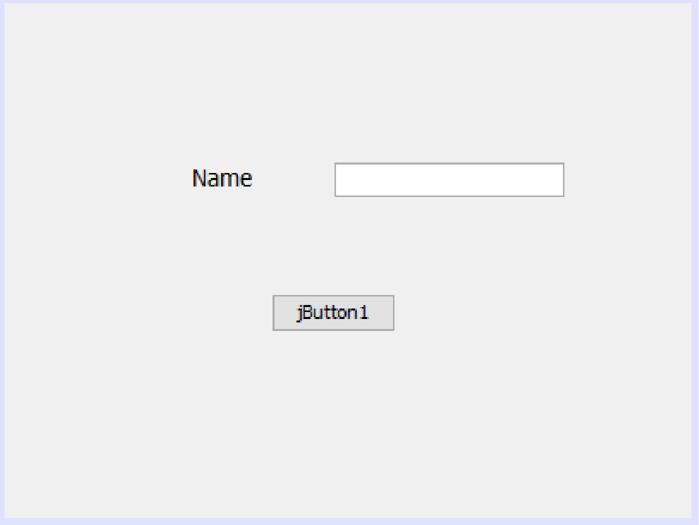
Give any project name to your project and then click on finish button.

Step 2:

Once the project is created, right click on project name and click New then click on JFrame form.

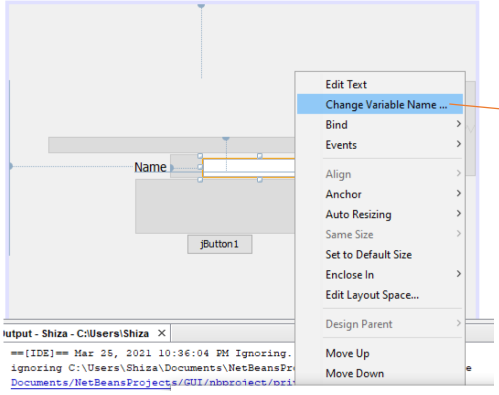
Step 3:

Add label, button and text field on frame. By right clicking on component change the variable name.

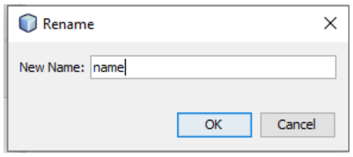


Step 4:

Change variable names by right clicking on component.



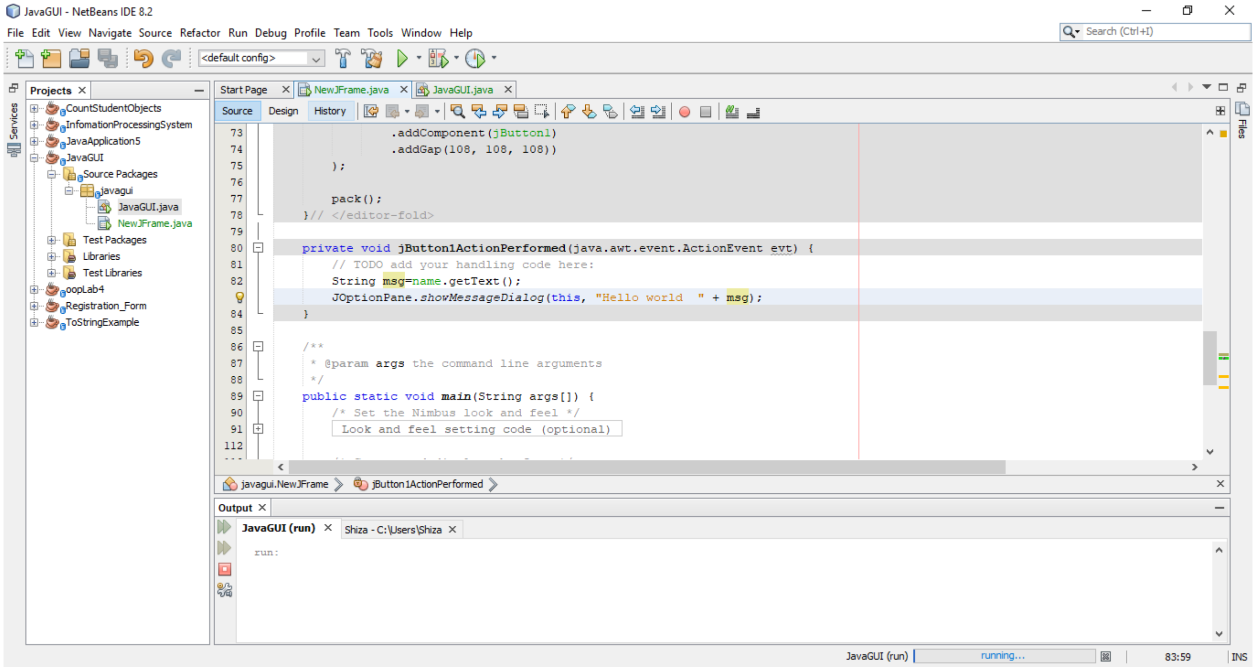
```
==[IDE]== Mar 25, 2021 10:36:04 PM Ignoring.
ignoring C:\Users\Shiza\Documents\NetBeansPr
Documents/NetBeansProjects/GUI/nbproject/pri
```



New Name: name

Step 5:

Double click on button you will see the jButton1ActionPerformed function. (This function will execute when jButton1ActionPerformed event will occur).



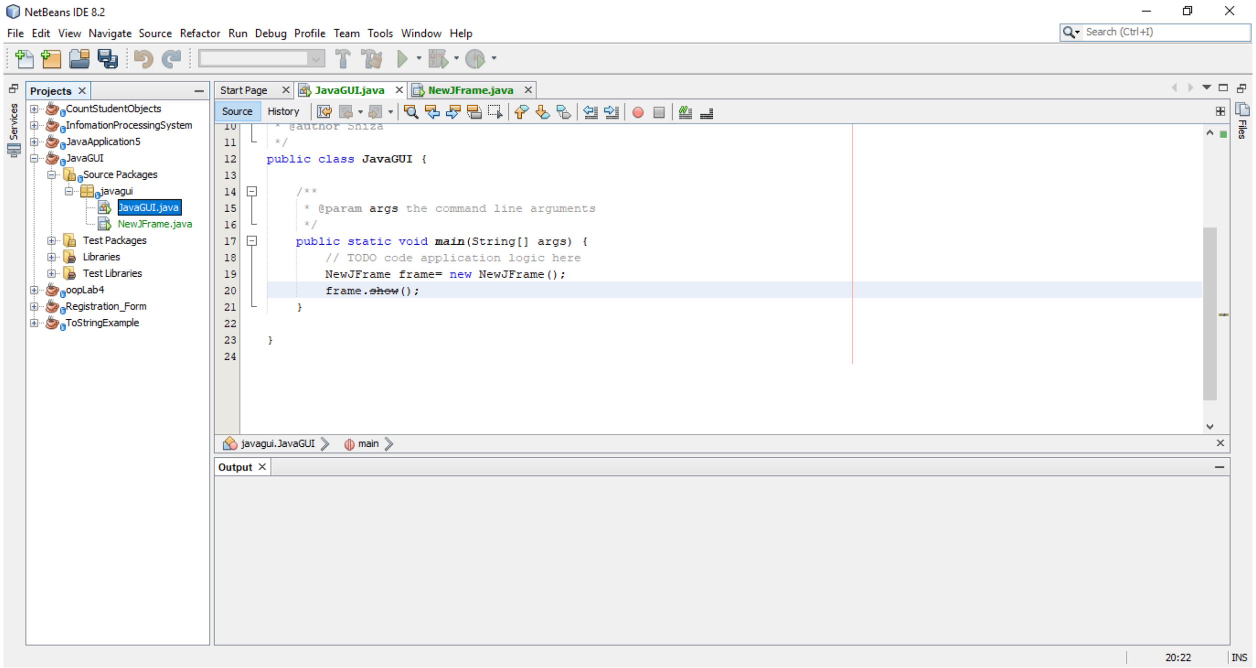
```
                .addComponent(jButton1)
                .addGap(108, 108, 108))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        String msg=name.getText();
        JOptionPane.showMessageDialog(this, "Hello world  " + msg);
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        Look and feel setting code (optional)
```

```
run:
```

Step 6:

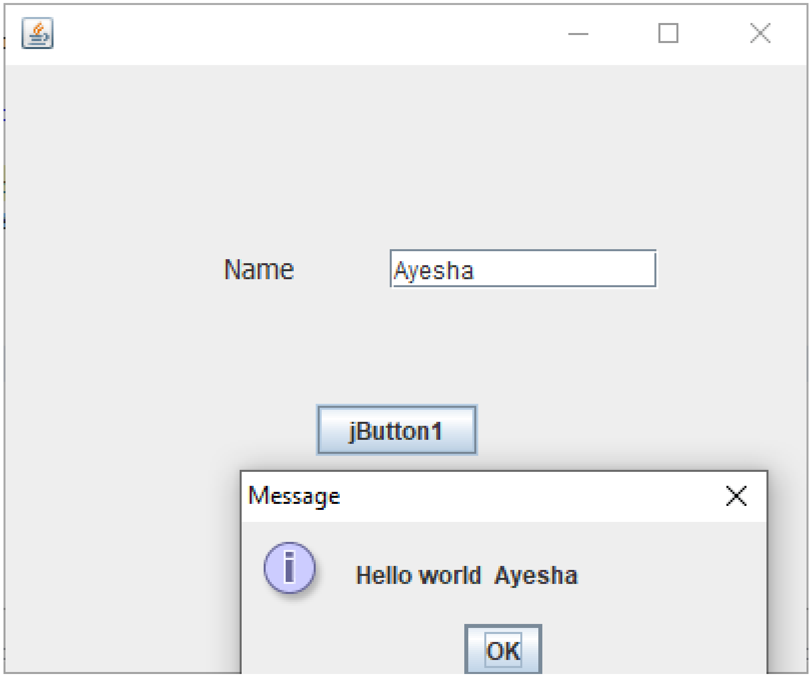
Call frame in main.

Output:



```
 * @author Shiza
 */
public class JavaGUI {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        NewJFrame frame= new NewJFrame();
        frame.show();
    }

}
```

2. A program that displays data in message box at button click

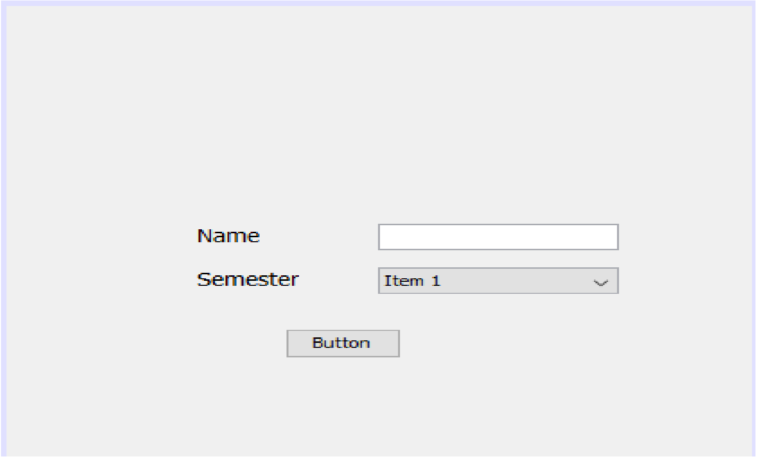


Name Ayesha

Message

Hello world Ayesha

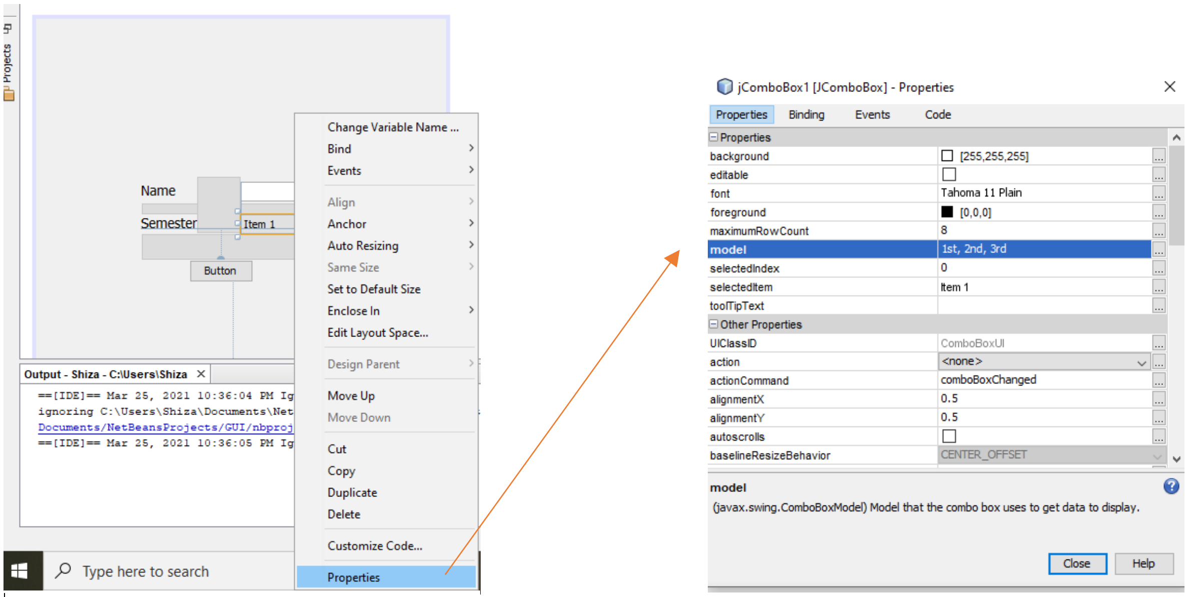
Step 1:
Create new project and new Jframe.
Step 2:
Add labels, text field, button & combo box on frame. And change variable names.



Name

Semester Item 1

Step 3:
Add items on combo box by right click and select properties option. Add required items in model.



Output - Shiza - C:\Users\Shiza

```
==[IDE]== Mar 25, 2021 10:36:04 PM Ig
ignoring C:\Users\Shiza\Documents\Net
Documents/NetBeansProjects/GUI/nbproj
==[IDE]== Mar 25, 2021 10:36:05 PM Ig
```

Properties

background	□ [255,255,255]
editable	□
font	Tahoma 11 Plain
foreground	■ [0,0,0]
maximumRowCount	8
model	1st, 2nd, 3rd
selectedIndex	0
selectedItem	Item 1
toolTipText	

Other Properties

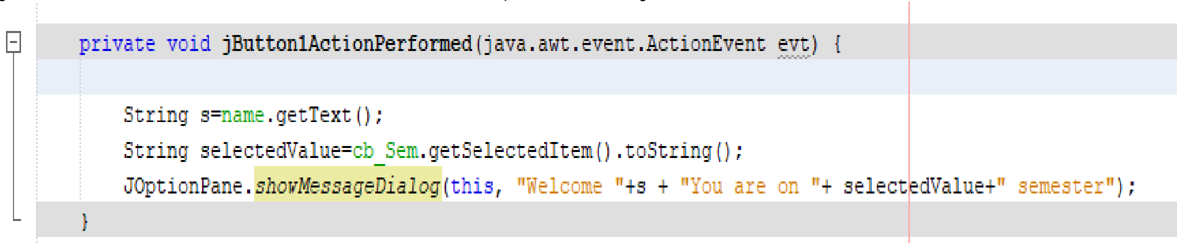
UIClassID	ComboBoxUI
action	<none>
actionCommand	comboBoxChanged
alignmentX	0.5
alignmentY	0.5
autoscrolls	□
baselineResizeBehavior	CENTER_OFFSET

model

(javax.swing.ComboBoxModel) Model that the combo box uses to get data to display.

Step 4:

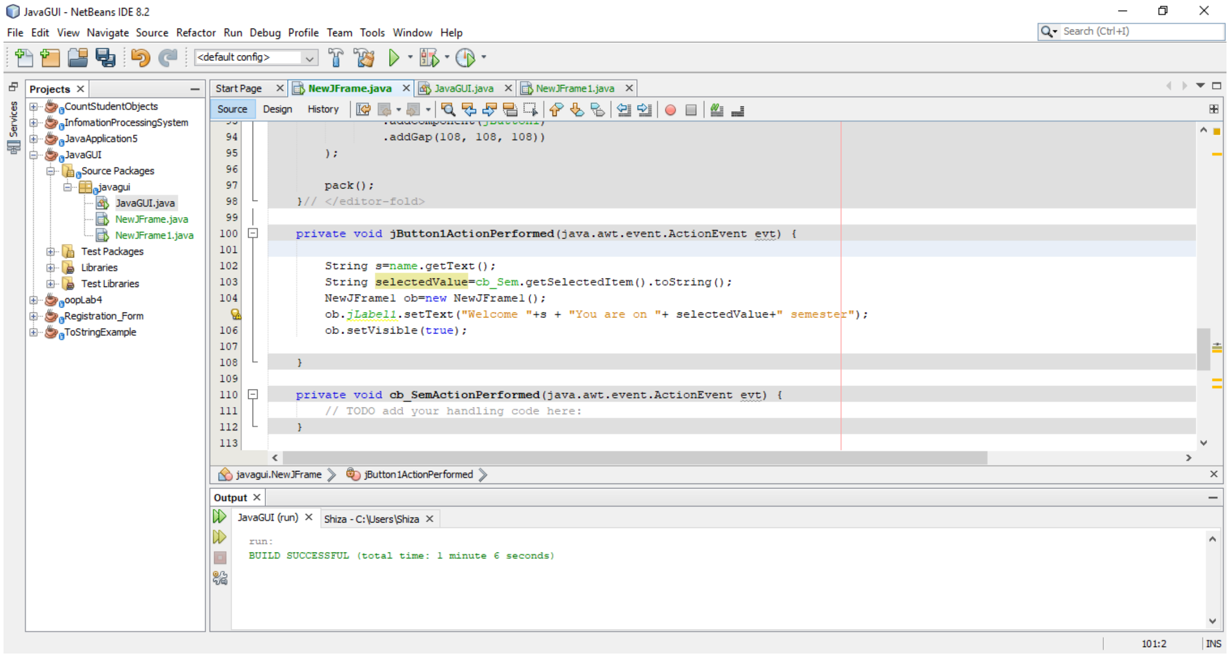
Double click on button you will see the jButton1ActionPerformed function. (This function will execute when jButton1ActionPerformed event will occur). Create object of other frame and set the label.



```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    String s=name.getText();
    String selectedValue=cb_Sem.getSelectedItem().toString();
    JOptionPane.showMessageDialog(this, "Welcome "+s + "You are on "+ selectedValue+" semester");
}
```

Step 7:

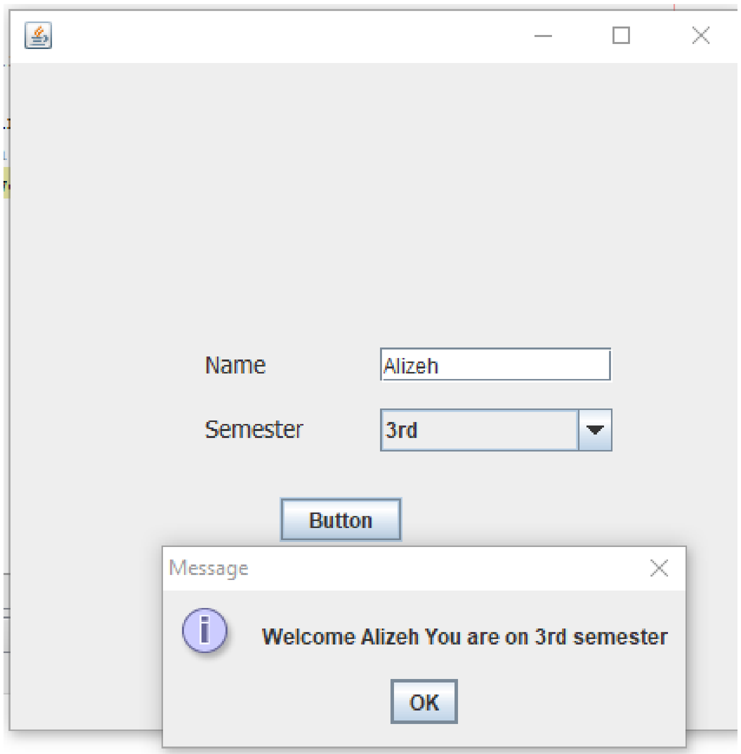
Call frame in main.



```
            .addGap(108, 108, 108))
    );

    pack();
}// </editor-fold>

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    String s=name.getText();
    String selectedValue=cb_Sem.getSelectedItem().toString();
    NewJFrame1 ob=new NewJFrame1();
    ob.jLabel1.setText("Welcome "+s + "You are on "+ selectedValue+" semester");
    ob.setVisible(true);

}

private void cb_SemActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}
```

```
run:
BUILD SUCCESSFUL (total time: 1 minute 6 seconds)
```

Output:



Name: Alizeh

Semester: 3rd

Message

Welcome Alizeh You are on 3rd semester

3. A program that sends data to another frame at button click.

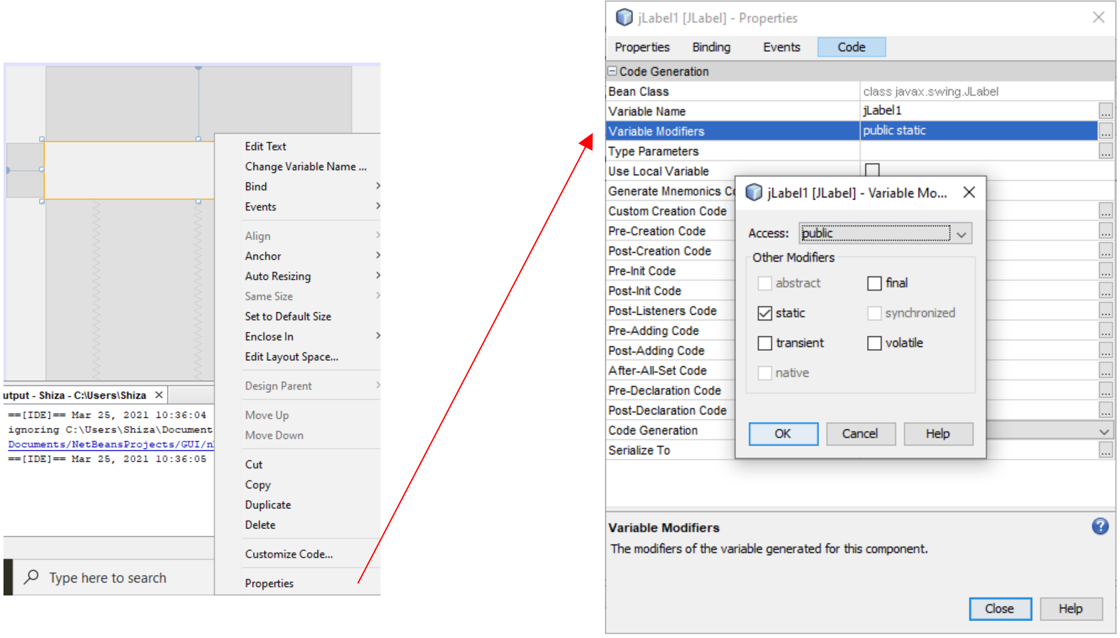
Step 1:
Create new project and new Jframe.

Step 2:
Add labels, text field, button & combo box on frame. And change variable names.

Step 3:
Add items on combo box by right click and select properties option. Add required items in model

Step 4:
Create another frame and add label.

Step 5:
For access label of second frame. We need change the access mode from private to public then it will accessible to other frame.After changing the mode we can access the text field of other frame.



```
utput - Shiza - C:\Users\Shiza
==[IDE]== Mar 25, 2021 10:36:04
ignoring C:\Users\Shiza\Document
Documents/NetBeansProjects/GUI/n
==[IDE]== Mar 25, 2021 10:36:05
```

jLabel1 [JLabel] - Properties

Bean Class	class javax.swing.JLabel
Variable Name	jLabel1
Variable Modifiers	public static
Type Parameters	
Use Local Variable	□
Generate Mnemonics C	
Custom Creation Code	
Pre-Creation Code	
Post-Creation Code	
Pre-Init Code	
Post-Init Code	
Post-Listeners Code	
Pre-Adding Code	
Post-Adding Code	
After-All-Set Code	
Pre-Declaration Code	
Post-Declaration Code	
Code Generation	
Serialize To	

jLabel1 [JLabel] - Variable Mo...

Access: public

Other Modifiers

□ abstract □ final

☑ static □ synchronized

□ transient □ volatile

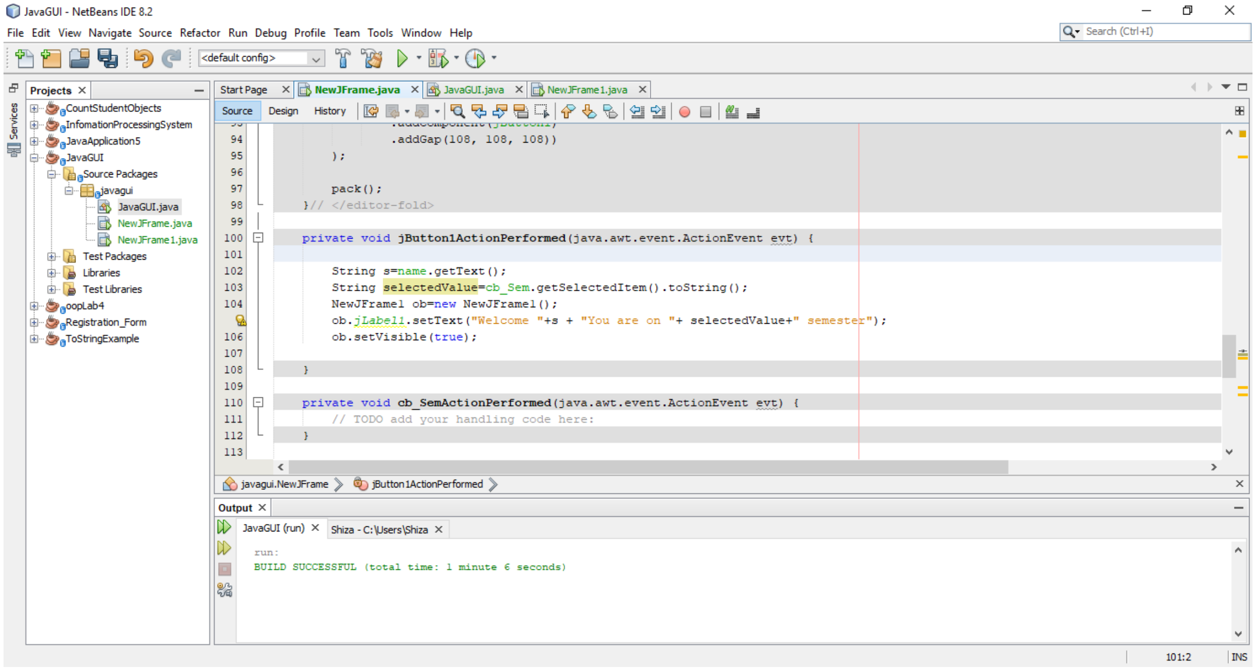
□ native

Variable Modifiers

The modifiers of the variable generated for this component.

Step 6:

Double click on button you will see the jButton1ActionPerformed function. (This function will execute when jButton1ActionPerformed event will occur). Create object of other frame and set the label.

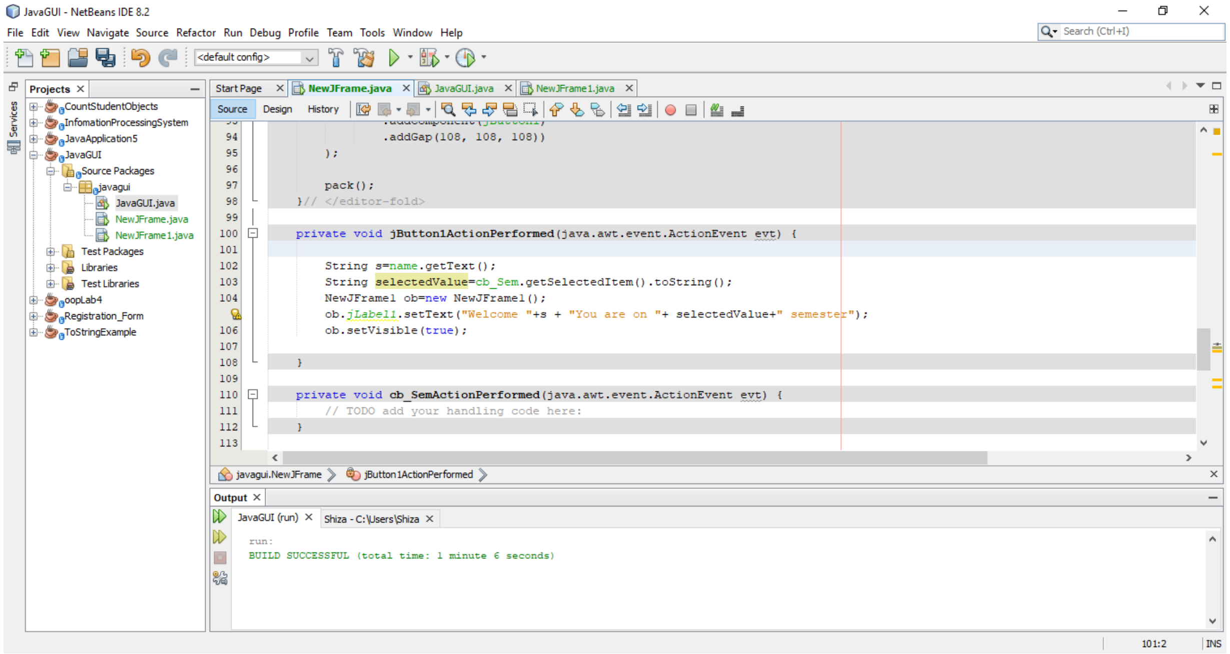


```
            .addGap(108, 108, 108))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        String s=name.getText();
        String selectedValue=cb_Sem.getSelectedItem().toString();
        NewJFrame1 ob=new NewJFrame1();
        ob.jLabel1.setText("Welcome "+s + "You are on "+ selectedValue+" semester");
        ob.setVisible(true);

    }

    private void cb_SemActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
```

```
run:
BUILD SUCCESSFUL (total time: 1 minute 6 seconds)
```

Step 7:

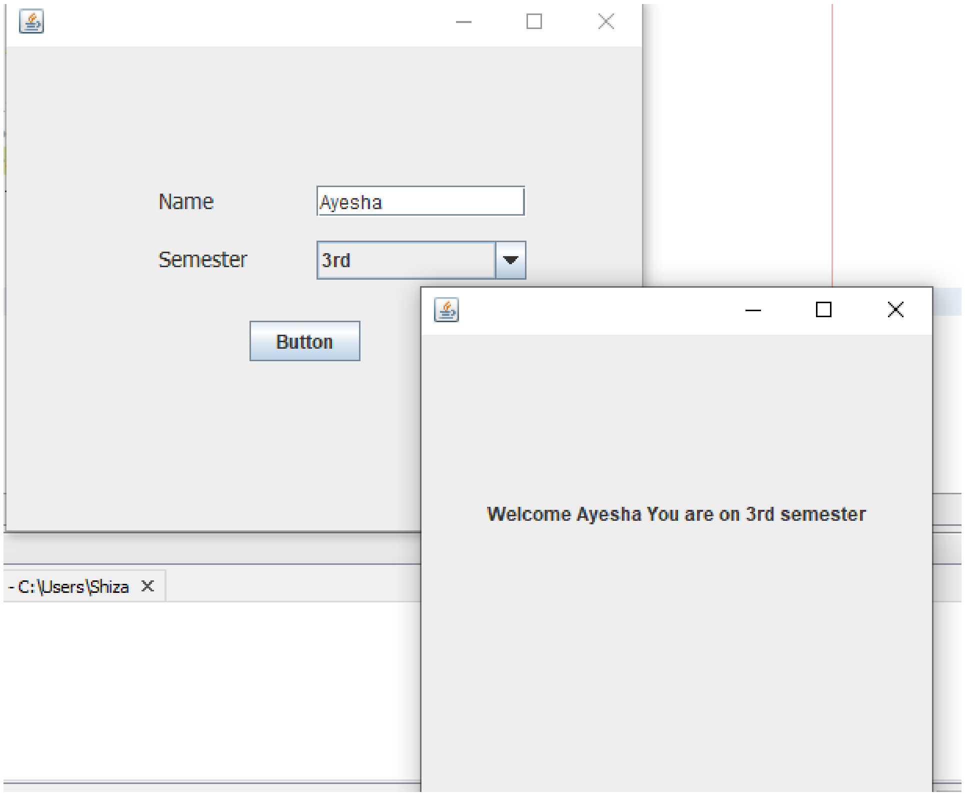
Call frame in main.



```
            .addGap(108, 108, 108))
        );

        pack();
    }// </editor-fold>

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

        String s=name.getText();
        String selectedValue=cb_Sem.getSelectedItem().toString();
        NewJFrame1 ob=new NewJFrame1();
        ob.jLabel1.setText("Welcome "+s + "You are on "+ selectedValue+" semester");
        ob.setVisible(true);

    }

    private void cb_SemActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
```

```
run:
BUILD SUCCESSFUL (total time: 1 minute 6 seconds)
```

Output:



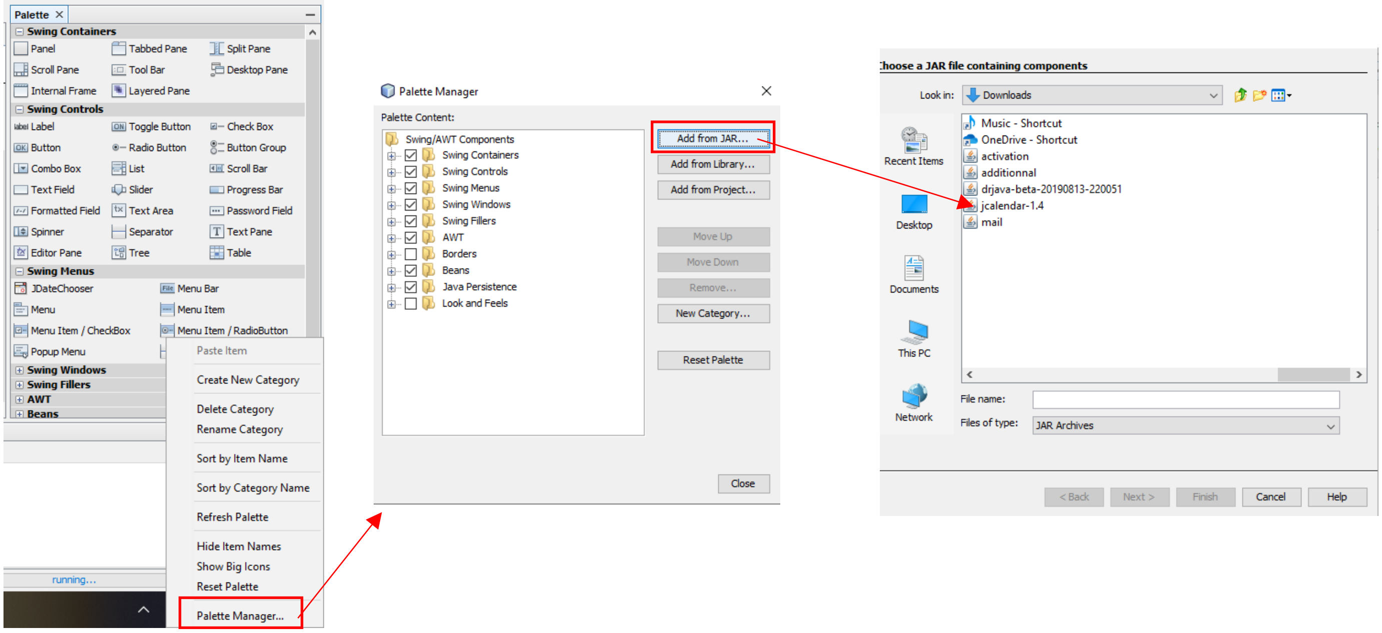
Name Ayesha

Semester 3rd

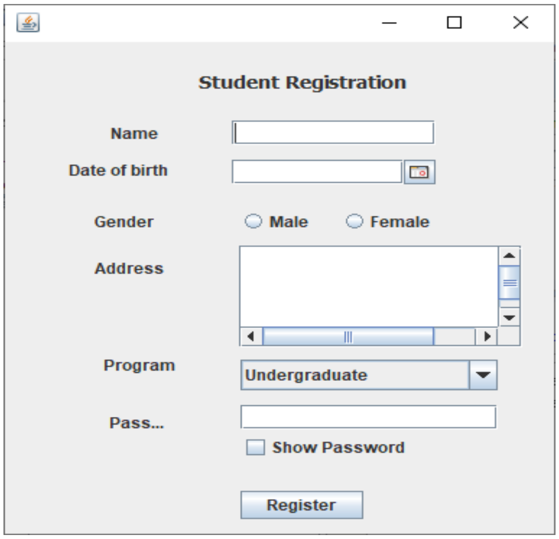
Welcome Ayesha You are on 3rd semester

4. **Make registration form that takes name of student, date of birth in calender format, gender in radio button, address in TextArea, Program in combo box, Password in Password field, also add check for show password.**

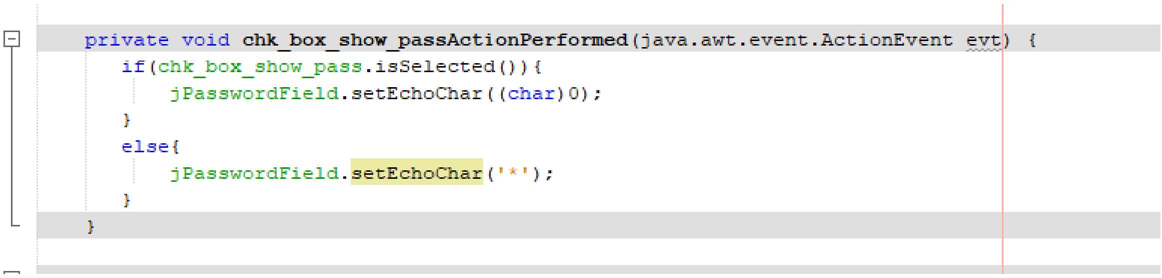
Step 1:
Create new Project on netbeans.
Step 2:
Once the project is created, right click on project name and click New then click on JFrame form.
Step 3:
Install JCalender.jar for JDateChooser from the following link.
http://www.java2s.com/example/jar/j/download-jcalendar14jar-file.html
After installing, right click on Palatte and select Palatte manager.



Step 4:
Form creation

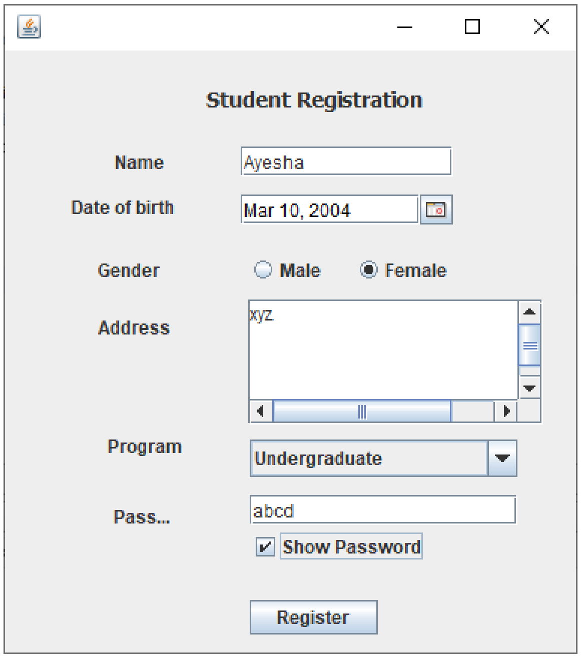


Step 5:
For show password, double click on check box, and write following into function.



```
private void chk_box_show_passActionPerformed(java.awt.event.ActionEvent evt) {
    if(chk_box_show_pass.isSelected()){
        jPasswordField.setEchoChar((char)0);
    }
    else{
        jPasswordField.setEchoChar('*');
    }
}
```

Output:



Student Registration

Name: Ayesha

Date of birth: Mar 10, 2004

Gender: ○ Male ● Female

Address: xyz

Program: Undergraduate

Pass...: abcd

☑ Show Password

Assignment

Students are required to develop employee attendance desktop application that keep records of employee with their attendance details. You are required to create an Attendance.java and Employee.java class.

Employee.java should have different employeeprofile data members and crud methods (create, retrieve, update, delete)

Attendance.java should have employee object, current datetime and status (A: Absent, P: Present, L: Leave) data member as private to manage attendance record

Driver.java should have list of attendance class object and maintain the record of attendance

The application will ask admin credentials for login (in driver class). Admin username and password should be "admin098". An admin can perform crud operations (add/delete/update) for employee and view attendance record based on view attendance record filter. The record filters include current date, specific date and specific month. To add new employee admin should enter EmployeeID (autogenerate according to previous existing record- EMP-X where X is natural number (1-n)), EmployeeName, EmailID, Designation, DOB, City and Gender. An admin can mark the attendance of registered employee. To keep record of attendance, create a file with name EmployeeRecord.txt that should have following records

.

```
EmpID, EmpName, DateTime, Status
EMP-1, Muhammad Ali, 26-03-2021 09:36:14, P
EMP-2, Fatima, 26-03-2021 12:40:24, A
```

****Students are required to use proper GUI form in this application**