

Exercise-17

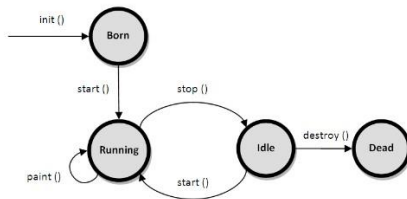
Applets**Q1. Define Applet.**

Ans: Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

Q2. List the applet life cycle methods.

Ans: It basically has five core methods namely:

init()
start()
stop()
paint()
destroy().

Q3. Draw the diagram showing different stages of applet life cycle. Ans:**Q4. What is the use of init method?**

Ans: Override the class initializer init() to initialize or allocate resources for the servlet instance's life, such as a counter. The init() method runs after the servlet is instantiated but before it accepts any requests.

Q5. What is the use of paint method?

Ans: The method paint() gives us access to an object of type Graphics class. Using the object of the Graphics class, we can call the drawString() method of the Graphics class to write a text message in the applet window.

Q6. How many constructors does applet have?

Ans: Applets don't usually have constructors because an applets isn't guaranteed to have a full environment until its init() method is called. For example, the Applet image loading methods don't work inside an applet constructor, but can be called from the init() method.

Q7. List the tools to execute applet program

Ans: There are two standard methods for running an applet:

By using Web Browser.

By using Applet Viewer

Q8. What is the use of applet tag in html file?

Ans: Applet tag in HTML file is used specify the applet class file, height and width of applet.

Q9. Write the html line of code that include applet in the html file. Ans: <applet code =

"Applet_File.class", height="h", width="w">

Q10. Does applet have file reading/writing capabilities.

Ans: . An applet cannot ordinarily read or write files on the execution host

.

Q11. What is the name of the package that supports applet? **Ans:** java.applet.* package supports applet.

Q12. Why applet are not support by java now?

Ans: java applets would depend on a Java Runtime Environment (JRE), a complex and heavy-weight software package. They also normally required a plug-in for the web browser.

Q13. From which version of java stop supporting applet?

Ans: Oracle announced in January 2016 that Applets would be deprecated in Java SE 9, and the technology was removed in Java SE 11.

PROGRAMS

Q1. Write a java program which overrides the following methods. After executing the Program: write when each method is called. init(), start(), paint(), stop and destroy.

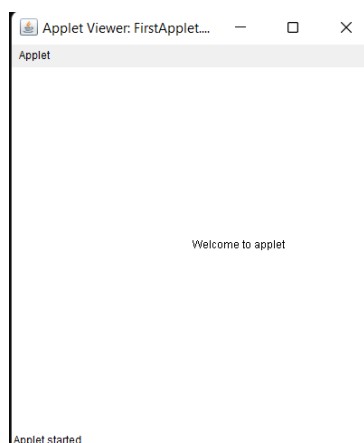
PROGRAM:

```
/*<Applet code="FirstApplet.class" width=600 height=600>
</Applet code>*/
import java.awt.*;
import java.applet.*;
public class FirstApplet extends Applet
{
    public void init()
    {
        System.out.println("INIT METHOD STARTED");
    }
    public void start()
    {
        System.out.println("START METHOD STARTED");
    }
    public void paint(Graphics G)
    {
        G.drawString("",1,1);
    }
    public void stop()
    {
        System.out.println("STOP METHOD STARTED");
    }
    public void destroy()
    {
        System.out.println("DESTROY METHOD STARTED");
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-106");
    }
}
```

OUTPUT:

```
A:\Java prg>javac FirstApplet.java

A:\Java prg>Appletviewer FirstApplet.java
Called only once
i will be called every time i get focused
Called after Paint
i lost focus
i will be called every time i get focused
Called after Paint
Called after Paint
Called after Paint
Called after Paint
Called after Paint
Called after Paint
i lost focus
its gone
THIS PROGRAM IS EXECUTED BY 21001-CS-106
```

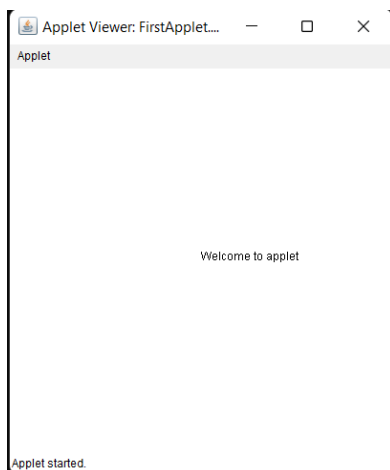


Q2. Write a java program to create an applet that displays welcome message.

PROGRAM:

```
/*<Applet code="SecondApplet.class" width=600 height=600>
</Applet code>*/
import java.awt.*;
import java.applet.*;
public class SecondApplet extends Applet
{
    public void init()
    {
        System.out.println("INIT METHOD STARTED");
    }
    public void start()
    {
        System.out.println("START METHOD STARTED");
    }
    public void paint(Graphics G)
    {
        G.drawString("Welcome",150,200);
    }
    public void stop()
    {
        System.out.println("STOP METHOD STARTED");
    }
    public void destroy()
    {
        System.out.println("DESTROY METHOD STARTED");
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-106");
    }
}
```

OUTPUT:



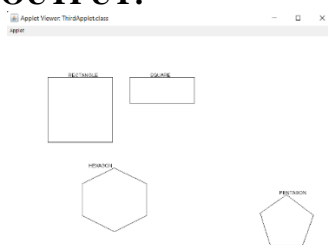
```
A:\Java prg>javac WelcomeApplet.java
A:\Java prg>Appletviewer WelcomeApplet.java
THIS PROGRAM IS EXECUTED BY 21001-CS-106
```

Q3. Write a java program to create an applet that displays square, rectangle, hexagoan, pentagon.

PROGRAM:

```
/*<Applet code="ThirdApplet.class" width=800 height=800>
</Applet code>*/
import java.awt.*;
import java.applet.*;
public class ThirdApplet extends Applet
{
    public void init()
    {
        System.out.println("INIT METHOD STARTED");
    }
    public void start()
    {
        System.out.println("START METHOD STARTED");
    }
    public void paint(Graphics G)
    {
        int a[]={262,341,341,262,184,184,262};
        int b[]={321,359,437,476,437,359,321};
        int c[]={682,748,721,643,618,682};
        int d[]={386,436,511,510,431,386};
        G.drawRect(300,100,158,63);
        G.drawString("RECTANGLE",150,100);
        G.drawRect(100,100,158,158);
        G.drawString("SQUARE",350,100);
        G.drawPolyline(a,b,7);
        G.drawString("HEXAGON",200,320);
        G.drawPolygon(c,d,6);
        G.drawString("PENTAGON",665,386);
    }
    public void stop()
    {
        System.out.println("STOP METHOD STARTED");
    }
    public void destroy()
    {
        System.out.println("DESTROY METHOD STARTED");
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-106");
    }
}
```

OUTPUT:



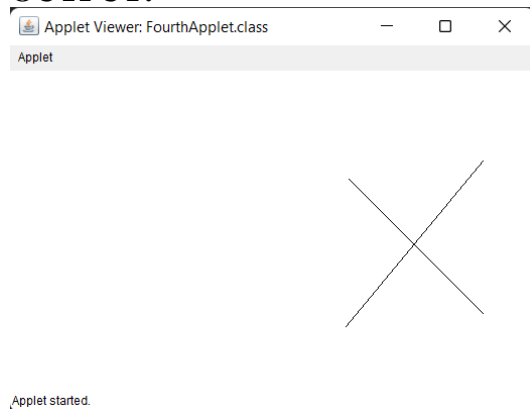
```
A:\Java Records>Appletviewer ThirdApplet.java
INIT METHOD STARTED
START METHOD STARTED
STOP METHOD STARTED
DESTROY METHOD STARTED
THIS PROGRAM WAS EXECUTED BY 21001-CS-106
```

Q4. Write a java program to create an applet that displays a line in the shape of “x”?

PROGRAM:

```
/*<Applet code="FourthApplet.class" width=600 height=600>
</Applet code>*/
import java.awt.*;
import java.applet.*;
public class FourthApplet extends Applet
{
    public void init()
    {
        System.out.println("INIT METHOD STARTED");
    }
    public void start()
    {
        System.out.println("START METHOD STARTED");
    }
    public void paint(Graphics G)
    {
        int[] a={ 336,469};
        int[] b={ 107,240};
        int[] c={ 469,333};
        int[] d={ 89,253};
        G.drawPolyline(a,b,2);
        G.drawPolyline(c,d,2);
    }
    public void stop()
    {
        System.out.println("STOP METHOD STARTED");
    }
    public void destroy()
    {
        System.out.println("DESTROY METHOD STARTED");
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-106");
    }
}
```

OUTPUT:



```
A:\Java Records>Appletviewer FourthApplet.java
INIT METHOD STARTED
START METHOD STARTED
STOP METHOD STARTED
DESTROY METHOD STARTED
THIS PROGRAM WAS EXECUTED BY 21001-CS-106
```

Q5. Write a java program to create an applet that displays snow man.

PROGRAM:

```
/*<Applet code="FifthApplet.class" width=900 height=900>
</Applet code>*/
import java.awt.*;
import java.applet.*;
public class FifthApplet extends Applet
{
    public void init()
    {
        System.out.println("INIT METHOD STARTED");
    }
    public void start()
    {
        System.out.println("START METHOD STARTED");
    }
    public void paint(Graphics G)
    {
        int a[]={455,600};
        int b[]={200,300};
        int c[]={275,150};
        int d[]={200,300};
        int e[]={355,375,335,355};
        int f[]={23,50,50,23};
        G.drawOval(300,50,117,111);
        G.drawOval(265,162,200,135);
        G.drawOval(230,297,280,160);
        G.drawOval(325,65,15,15);
        G.drawOval(380,65,15,15);
        G.drawPolyline(a,b,2);
        G.drawPolyline(c,d,2);
        G.drawPolyline(e,f,4);
        G.drawOval(353,210,15,15);
        G.drawOval(353,245,15,15);
        G.drawOval(353,330,15,15);
        G.drawOval(353,375,15,15);
        G.drawOval(353,420,15,15);
    }
    public void stop()
    {
        System.out.println("STOP METHOD STARTED");
    }
    public void destroy()
    {
        System.out.println("DESTROY METHOD STARTED");
    }
}
```

OUTPUT:



Exercise-18
Layout Managers

Q1. What is a Graphical user interface?

Ans: GUI, which stands for Graphical User Interface, is a user-friendly visual experience builder for Java applications. It comprises graphical units like buttons, labels, windows, etc. via which users can connect with an application. Swing and JavaFX are two commonly used applications to create GUIs in Java.

Q2. What is the advantage of GUI over CUI?

Ans: The main difference between GUI and CUI is that in GUI, the user interacts with a computer using graphics like images and icons, while in CUI, the user interacts with a computer using commands like text. A user interacts with applications by using graphics.

Q3. What is a layout Manager?

Ans: A layout manager is an object that implements the `LayoutManager` interface* and determines the size and position of the components within a container. Although components can provide size and alignment hints, a container's layout manager has the final say on the size and position of the components within the container.

Q4. List five layout Managers.?

Ans: List of Layout Managers

`java.awt.BorderLayout.`
`java.awt.FlowLayout.`
`java.awt.CardLayout.`
`java.awt.GridLayout.`
`java.awt.GridBagLayout.`

Q5. List the constants to place the components in the Border Layout Manager?

Ans: Constant border layouts are the constraint must be one of the following constants:

- 1.NORTH
- 2.SOUTH
- 3.EAST
- 4.WEST
- 5.CENTER

Q6. List the three constructors of Flow Layout.

Ans: `FlowLayout()`
`FlowLayout(int align)`
`FlowLayout(int align, int hgap, int vgap)`

Q7. What is the default layout manager of Frame?

Ans: The `BorderLayout` manager is the default Layout Manager for Frames .

Q8. What is the default layout manager of Applet?

Ans: `FlowLayout`. The flow layout is the default layout manager for all `Panel` objects and applets.

Q9. Write the procedure to place components in card Layout **Ans:** Procedure to place components in Card Layout:

- ☐ Create a Card Layout object.
- ☐ Create a container to hold the components.
- ☐ Add each component to the container with a unique identifier.

☐

Add the container to a panel or frame.

Q10. Write the procedure to place components in GridBag Layout? Ans: Procedure to place components in GridBag Layout :

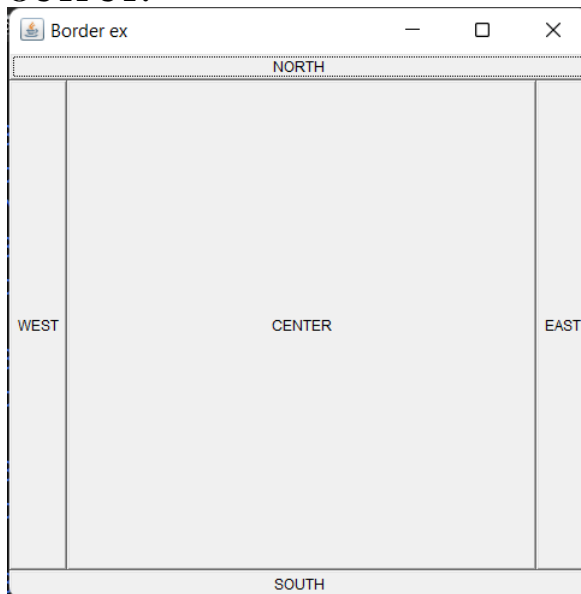
- ☐ Create a new GridBagLayout object.
- ☐ Create a container to hold the components that will be displayed in the layout.
- ☐ Create a GridBagConstraints class object for adding components to the layout.
- ☐ Set the constraints for each component by specifying its grid location, size, anchor, and fill.
- ☐ Add each component to the container using the add() method and its associated GridBagConstraints object.
- ☐ Add the container to a panel or frame to display the layout in a window.

PROGRAMS

Q1. Write a java program to design a GUI using Border Layout.

PROGRAM:

```
import java.awt.*;
class BorderExample
{
    Button b1,b2,b3,b4,b5;
    Frame f1;
    public BorderExample()
    {
        f1=new Frame("Border ex");
        b1=new Button("NORTH");
        b2=new Button("SOUTH");
        b3=new Button("EAST");
        b4=new Button("WEST");
        b5=new Button("CENTER");
        f1.setSize(500,500);
        f1.setLayout(new BorderLayout());
        f1.add(b1,BorderLayout.NORTH);
        f1.add(b2,BorderLayout.SOUTH);
        f1.add(b3,BorderLayout.EAST);
        f1.add(b4,BorderLayout.WEST);
        f1.add(b5,BorderLayout.CENTER);
        f1.setVisible(true);
    }
    public static void main(String s[])
    {
        BorderExample bx=new BorderExample();
        System.out.println("This Program Is Executed By 21001-CS-106");
    }
}
```

OUTPUT:

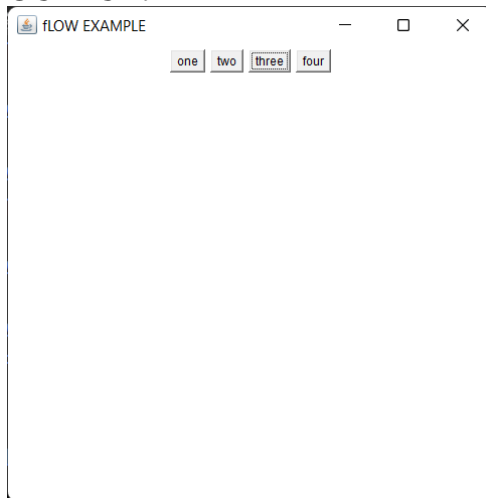
```
A:\Java Records>javac BorderExample.java
A:\Java Records>java BorderExample
This Program Is Executed By 21001-CS-106
```

Q2. Write a java program to design a GUI using Flow Layout.

PROGRAM:

```
import java.awt.*;
class FlowExample
{
    Button b1,b2,b3,b4;
    Frame f1;
    FlowEx()
    {
        f1=new Frame("fLOW EXAMPLE");
        f1.setLayout(new FlowLayout());
        f1.setSize(500,500);
        b1= new Button("one");
        b2= new Button("two");
        b3= new Button("three");
        b4= new Button("four");
        f1.add(b1);
        f1.add(b2);
        f1.add(b3);
        f1.add(b4);
        f1.setVisible(true);
    }
    public static void main(String s[])
    {
        FlowEx fe=new FlowEx();
        System.out.println("This Program Is Executed By 21001-CS-106");
    }
}
```

OUTPUT:



```
A:\Java Records>javac FlowExample.java

A:\Java Records>java FlowExample
This Program Is Executed By 21001-CS-106
```

Q3. Write a java program to design a GUI using Card Layout consisting of five child panel and each panel consist of a label and Text field in the first line and button in the next line. Create a list of Buttons to navigate through the panels.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
class CardLayoutEx extends Frame implements ActionListener
{
    Button first , prv,next,last;
    Panel parent;
    Panel C1,C2,C3,C4,C5;
    Label cl1,cl2,cl3,cl4,cl5;
    TextField C11,C12,C13,C14,C15;
    CardLayout cardlayout ;
    Button f ;
    Button b;
    public CardLayoutEx()
    {
        next= new Button("Shiv");
        prv= new Button("Prakash");
        first= new Button("Aakash");
        last= new Button("Charan");
        b= new Button("Ajith");
        f= new Button("Navigate to next ->");

        next.addActionListener(this);
        first.addActionListener(this);
        last.addActionListener(this);
        prv.addActionListener(this);
        b.addActionListener(this);

        f.addActionListener(this);

        C1=new Panel();
        cl1=new Label("White Text");
        C11=new TextField("Black Text");
        C1.setLayout(new FlowLayout());
        C1.add(cl1);
        C1.add(C11);

        C2=new Panel();
        cl2=new Label("orange Text");
        C12=new TextField("Blue Text");
        C2.setLayout(new FlowLayout());
```

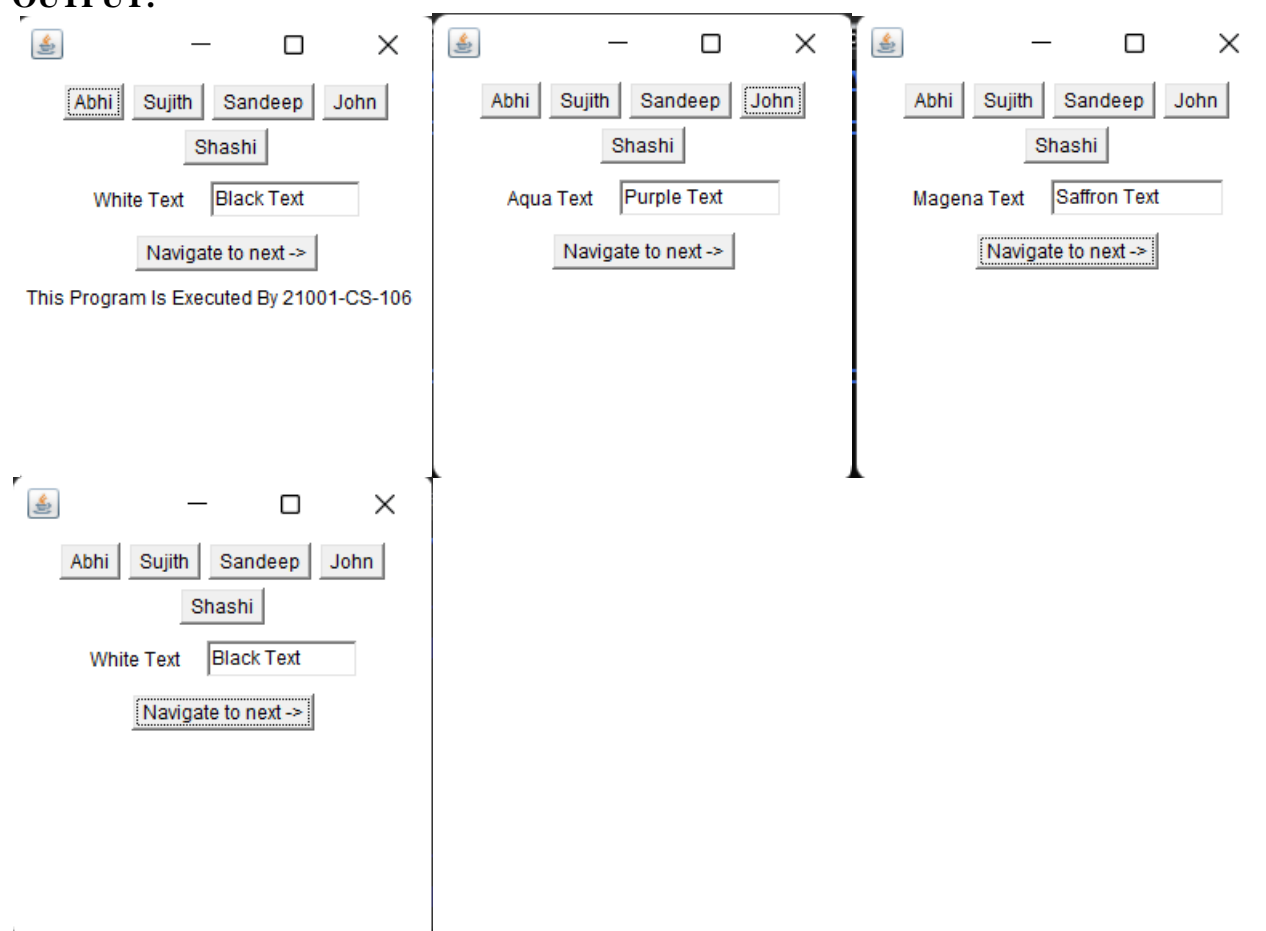
```
C2.add(cl2);
C2.add(Cl2);
C3=new Panel();
cl3=new Label("red Text");
Cl3=new TextField("pink Text");
C3.setLayout(new FlowLayout());
C3.add(cl3);
C3.add(Cl3);
C4=new Panel();
cl4=new Label("Aqua Text");
Cl4=new TextField("Purple Text");
C4.setLayout(new FlowLayout());
C4.add(cl4);
C4.add(Cl4);

C5=new Panel();
cl5=new Label("Magenta Text");
Cl5=new TextField("Saffron Text");
C5.setLayout(new FlowLayout());
C5.add(cl5);
C5.add(Cl5);
parent = new Panel();
cardlayout = new CardLayout();
parent.setLayout(cardlayout); // need to pass the created object not a new
parent.add(C1,"First");
parent.add(C2,"Second");
parent.add(C3,"Third");
parent.add(C4,"Fourth");
parent.add(C5,"Fifth");
setLayout(new FlowLayout());
add(next);
add(prv);
add(first);
add(last);
add(b);
add(parent);
add(f);
setSize(280,300);
setVisible(true);
}
public void actionPerformed(ActionEvent e)
{
    if(e.getActionCommand()=="Shiv")
        cardlayout.first(parent);
    else if(e.getActionCommand()=="Prakash")
        cardlayout.previous(parent);
    else if(e.getActionCommand() == "Aakash")
```

```

        cardlayout.last(parent);
    else if(e.getActionCommand() == "Charan")
        cardlayout.next(parent);
    else if(e.getActionCommand() == "Ajith")
        cardlayout.previous(parent);
    else
        cardlayout.next(parent);
}
public static void main(String[] args)
{
    CardLayoutEx obj = new CardLayoutEx();
}
}

```

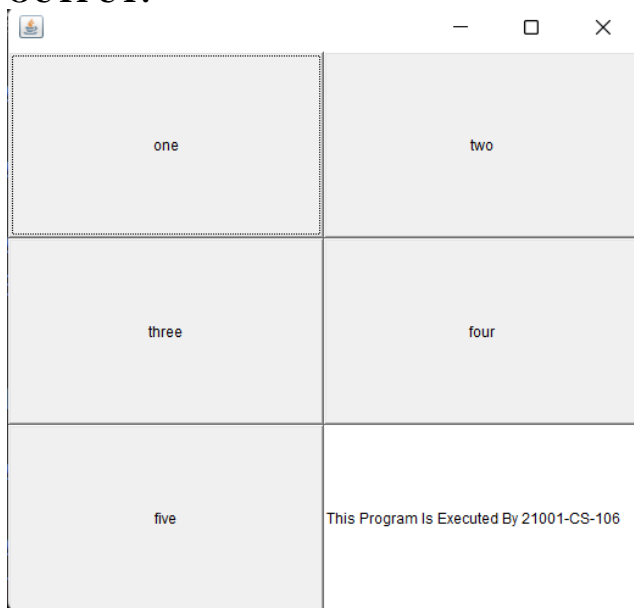
OUTPUT:

Q4. Write a java program to design a GUI using Grid Layout.

PROGRAM:

```
import java.awt.*;
class GridEx extends Frame
{
    Button b1,b2,b3,b4,b5;
    Label name;
    public GridEx()
    {
        b1= new Button("one");
        b2= new Button("two");
        b3= new Button("three");
        b4= new Button("four");
        b5= new Button("five");
        name=new Label("This Program Is Executed By 21001-CS-106")
        setSize(500,500);
        setLayout(new GridLayout(3,3));
        add(b1);
        add(b2);
        add(b3);
        add(b4);
        add(b5);
        setVisible(true);
    }
    public static void main(String s[])
    {
        GridEx g=new GridEx();
    }
}
```

OUTPUT:



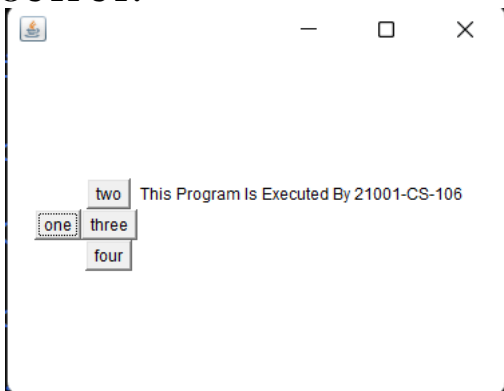
Q5. Write a java program to design a GUI using GridBag Layout.

PROGRAM:

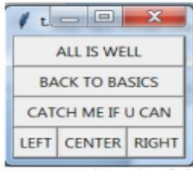
```
import java.awt.*;
class GridBagEx extends Frame
{
    Button b1,b2,b3,b4;
    GridBagLayout gbl;
    Label name=new Label("This Program Is Executed By 21001-CS-106");
    GridBagConstraints gbc;
    public GridBagEx()
    {
        b1=new Button("one");
        b2=new Button("two");
        b3=new Button("three");
        b4=new Button("four");
        gbl=new GridBagLayout();
        gbc=new GridBagConstraints();
        setLayout(gbl);
        gbc.gridx=0;
        gbc.gridy=1;
        add(b1,gbc);
        gbc.gridx=1;
        gbc.gridy=0;
        add(b2,gbc);
        gbc.gridx=1;
        gbc.gridy=1;
        add(b3,gbc);
        gbc.gridx=1;
        gbc.gridy=2;
        add(b4,gbc);
        add(name);
        setSize(300,300);
        setVisible(true);
    }

    public static void main(String s[])
    {
        GridBagEx gx=new GridBagEx();
    }
}
```

OUTPUT:



Q6. Write a java program to create the following GUI



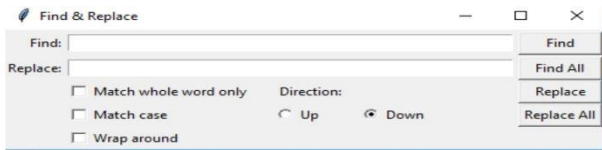
PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
class GridBagLayoutExample2 extends Frame
{
    Button b1,b2,b3,b4,b5,b6;
    public GridBagLayoutExample2()
    {
        setLayout(new GridBagLayout());
        Label name=new Label("This Program Is Executed By 21001-CS-106");
        GridBagConstraints gs=new GridBagConstraints();
        b1=new Button("    ALL IS WELL    ");
        b2=new Button("    BACK TO BASICS    ");
        b3=new Button(" CATCH ME IF YOU CAN ");
        b4=new Button("LEFT");
        b5=new Button("CENTER");
        b6=new Button("RIGHT");
        gs.gridwidth=1;
        gs.gridx=1;
        gs.gridy=3;
        add(b4,gs);
        gs.gridwidth=1;
        gs.gridx=2;
        gs.gridy=3;
        add(b5,gs);
        gs.gridwidth=1;
        gs.gridx=3;
        gs.gridy=3;
        add(b6,gs);
        gs.gridwidth=3;
        gs.gridx=1;
        gs.gridy=0;
        add(b1,gs);
        gs.gridwidth=3;
        gs.gridx=1;
        gs.gridy=1;
        add(b2,gs);
        gs.gridwidth=3;
        gs.gridx=1;
        gs.gridy=2;
        add(b3,gs);
        add(name);
        setSize(300,300);
    }
}
```

```
setVisible(true);  
}  
public static void main(String s[])  
{  
    new GridBagLayoutExample2();  
}  
}
```

OUTPUT:

Q7. Write a java program to create the following GUI.



PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
class GridBagLayoutExample extends Frame
{
    Checkbox c1,c2,c3,c4,c5;
    Button b1,b2,b3,b4;
    Label l1,l2,l3;
    TextField t1,t2;
    GridBagLayout g;
    public GridBagLayoutExample()
    {
        g=new GridBagLayout();
        GridBagConstraints gx=new GridBagConstraints();
        setLayout(g);
        Label name=new Label("This Program Is Executed By 21001-CS-106");
        CheckboxGroup cb=new CheckboxGroup();
        c1=new Checkbox("Match whole word only");
        c2=new Checkbox("Match case");
        c3=new Checkbox("Wrap around");
        c4=new Checkbox("Up",cb,false);
        c5=new Checkbox("Down",cb,false);
        b1=new Button(" Find ");
        b2=new Button(" Find All "); b3=new Button(" Replace ");
        b4=new Button("Replace All");
        l1=new Label("Find:");
        l2=new Label("Replace");
        l3=new Label("Direction");
        t1=new TextField(70);
        t2=new TextField(70);
        gx.gridx=1;
        gx.gridy=2;
        add(c1,gx);
        gx.gridx=1;
        gx.gridy=3;
        add(c2,gx);
        gx.gridx=1;
        gx.gridy=4;
        add(c3,gx);
        gx.gridx=3;
        gx.gridy=2;
        add(l3,gx);
        gx.gridx=3;
        gx.gridy=3;
```

```
        add(c4,gx);
        gx.gridx=4;
        gx.gridy=3;
        add(c5,gx);
        gx.gridx=0;
        gx.gridy=0;
        add(l1,gx);
        gx.gridx=0;
        gx.gridy=1;
        add(l2,gx);
        gx.gridwidth=5;
        gx.gridx=1;
        gx.gridy=0;
        add(t1,gx);
        gx.gridx=9;
        gx.gridy=0;
        add(b1,gx);
        gx.gridx=9;
        gx.gridy=1;
        add(b2,gx);
        gx.gridx=9;
        gx.gridy=2;
        add(b3,gx);
        gx.gridx=9;
        gx.gridy=3;
        add(b4,gx);
        gx.gridx=1;
        gx.gridy=1;
        add(t2,gx);
        add(name);
        pack();
        setSize(300,300);
        setVisible(true);
        pack();
    }
    public static void main(String s[])
    {
        GridBagLayoutExample ggg=new GridBagLayoutExample();
    }
}
```

OUTPUT:

Exercise-19 Event Handling

Q1. Define component?

Ans: A component is an object having a graphical representation that can be displayed on the screen and that can interact with the user. Examples of components are the buttons, checkboxes, and scrollbars of a typical graphical user interface.

Q2. Define container.

Ans: Containers are the interface between a component and the low-level, platform-specific functionality that supports the component. Before it can be executed, a web, enterprise bean, or application client component must be assembled into a Java EE module and deployed into its container.

Q3. Is container is a component?

Ans: yes a container is component.

Q4. Is component is a container?

Ans: No all component are not container.

Q5. List four containers?

Ans: Frame, Panel, and Applet and Dialog

Q6. List ten components which are not containers?

Ans: Ten components which are not containers are :

- | | |
|--------------|-------------------|
| 1. Button | 6. Label |
| 2. Scrollbar | 7. Menubar |
| 3. Checkbox | 8. TextField |
| 4. TextArea | 9. List |
| 5. Choice | 10. CheckboxGroup |

Q7. Define Event Delegation Model?

Ans: The delegation event model in Java is a way of handling events that involve three objects: the source of the event, the listener that is notified of the event, and the event object that contains information about the event.

Q8. List five Event Classes.

Ans:

ActionEvent.
AdjustmentEvent. • ComponentEvent.
ContainerEvent.
FocusEvent.
InputEvent.
ItemEvent.
KeyEvent.

Q9. List five Event Listeners.

Ans: Five Event Listeners :

1. ActionListener
2. ItemListener
3. WindowListener
4. AdjustmentListener

5.KeyListener

Q10. List five Event Sources. Ans:Five Event Sources :

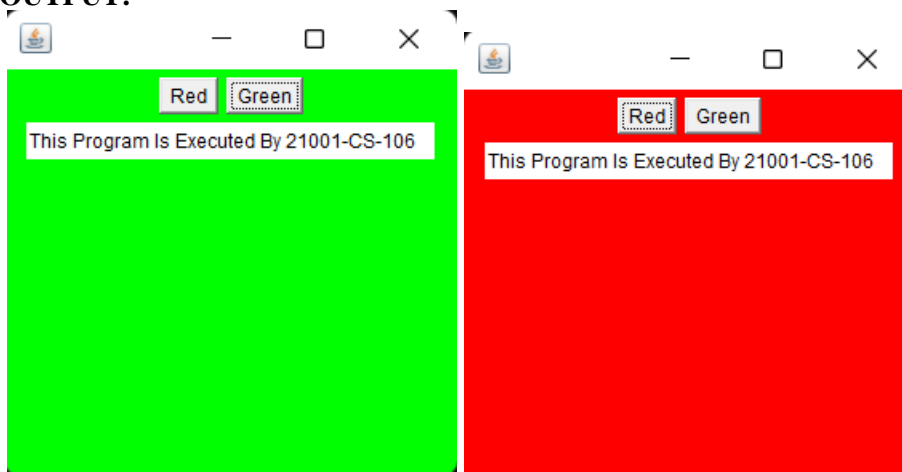
- 1.Buttons 2.Checkboxes,
- 3.List 4.Choice
- 5.Scrollbar

PROGRAMS

Q1. Write a java program to handle ActionEvent generated by Button.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
class ButtonEvent extends Frame implements ActionListener
{
    Button b1,b2;
    Label name=new Label("This Program Is Executed By 21001-CS-106");
    public ButtonEvent()
    {
        setLayout(new FlowLayout());
        b1=new Button("Red");
        b2=new Button("Green");
        add(b1);
        add(b2);
        b1.addActionListener(this);
        b2.addActionListener(this);
        add(name);
        setSize(300,300);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getActionCommand()=="Red")
            setBackground(Color.red);
        if(e.getActionCommand()=="Green")
            setBackground(Color.green);
    }
    public static void main(String s[])
    {
        new ButtonEvent();
    }
}
```

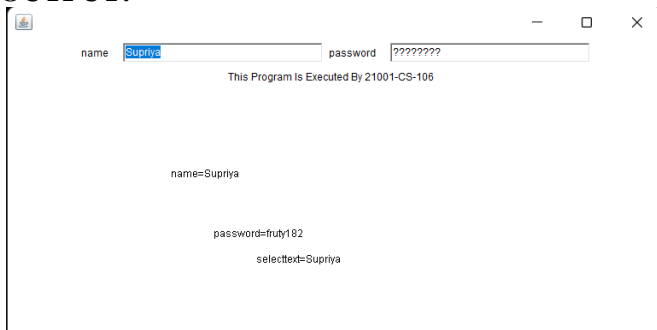
OUTPUT:

Q2. Write a java program to handle ActionEvent generated by TextField

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
public class TextEx extends Frame implements ActionListener
{
    Label l1,l2;
    TextField t1,t2;
    Label name=new Label("This Program Is Executed By 21001-CS-106");
    public TextEx()
    {
        l1=new Label("name");
        l2=new Label("password");
        t1=new TextField(30);
        t2=new TextField(30);
        setLayout(new FlowLayout());
        t2.setEchoChar('?');
        add(l1);
        add(t1);
        add(l2);
        add(t2);
        add(name);
        t1.addActionListener(this);
        t2.addActionListener(this);
        setSize(400,400);
        setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString("name"+t1.getText(),200,200);
        g.drawString("password"+t2.getText(),250,270);
        g.drawString("selecttext"+t1.getSelectedText(),300,300);
    }
    public static void main(String s[])
    {
        TextEx te=new TextEx();
    }
}
```

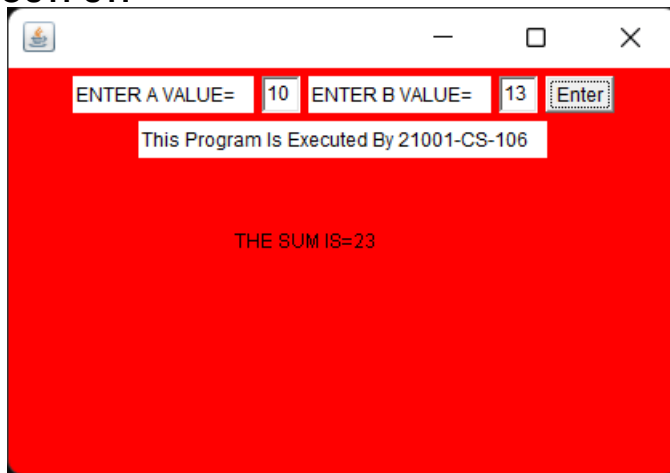
OUTPUT:



Q3. Write a java program to with a Button and Two Text Fields, When Button is pressed the numbers entered in the text field are summed and their sum is displayed on the label.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
class Q3Text extends Frame implements ActionListener
{
    int num1,num2,res;
    Button b1;
    Label l1,l2;
    TextField t1,t2;
    FlowLayout p1;
    Label name=new Label("This Program Is Executed By 21001-CS-106");
    public Q3Text()
    {
        p1=new FlowLayout();
        setLayout(p1);
        l1=new Label("ENTER A VALUE=");
        l2=new Label("ENTER B VALUE=");
        t1=new TextField();
        t2=new TextField();
        b1=new Button("Enter");
        add(l1);
        add(t1);
        add(l2);
        add(t2);
        add(b1);
        add(name);
        setSize(300,300);
        setVisible(true);
        b1.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getActionCommand().equals("SUBMIT"))
        {
            setBackground(Color.green);
            num1=Integer.parseInt(t1.getText());
            num2=Integer.parseInt(t2.getText());
            res=num1+num 2;
            repaint();
        }
    }
    public void paint(Graphics g)
    {
        g.drawString("THE SUM IS="+res,150,150);
    }
    public static void main(String s[])
    {
        new Q3Text();
    }
}
```

OUTPUT:

ENTER A VALUE= 10 ENTER B VALUE= 13 Enter

This Program Is Executed By 21001-CS-106

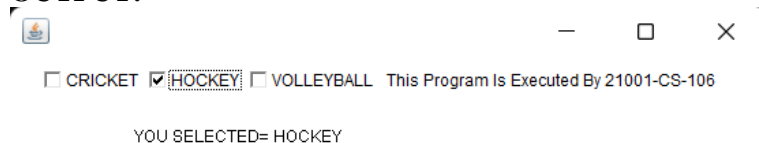
THE SUM IS=23

Q4. Write a java program to handle ItemStateChangedEvent generated by Checkboxes

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
class EventCheckbox extends Frame implements ItemListener
{
    Checkbox c1,c2,c3;
    String k="";
    Label name=new Label("This Program Is Executed By 21001-CS-106");
    public EventCheckbox()
    {
        setLayout(new FlowLayout());
        c1=new Checkbox("CRICKET");
        c2=new Checkbox("HOCKEY");
        c3=new Checkbox("VOLLEYBALL");
        add(c1);
        add(c2);
        add(c3);
        add(name);
        c1.addItemListener(this);
        c2.addItemListener(this);
        c3.addItemListener(this);
        setSize(300,300);
        setVisible(true);
    }
    public void itemStateChanged(ItemEvent e)
    {
        if(c1.getState()==true)
            k=k+c1.getLabel();
        if(c2.getState()==true)
            k=k+c2.getLabel();
        if(c3.getState()==true)
            k=k+c3.getLabel();
        repaint();
    }
    public void paint(Graphics G)
    {
        G.drawString("YOU SELECTED="+k,100,100);
        k="";
    }
    public static void main(String s[])
    {
        new EventCheckbox();
    }
}
```

OUTPUT:

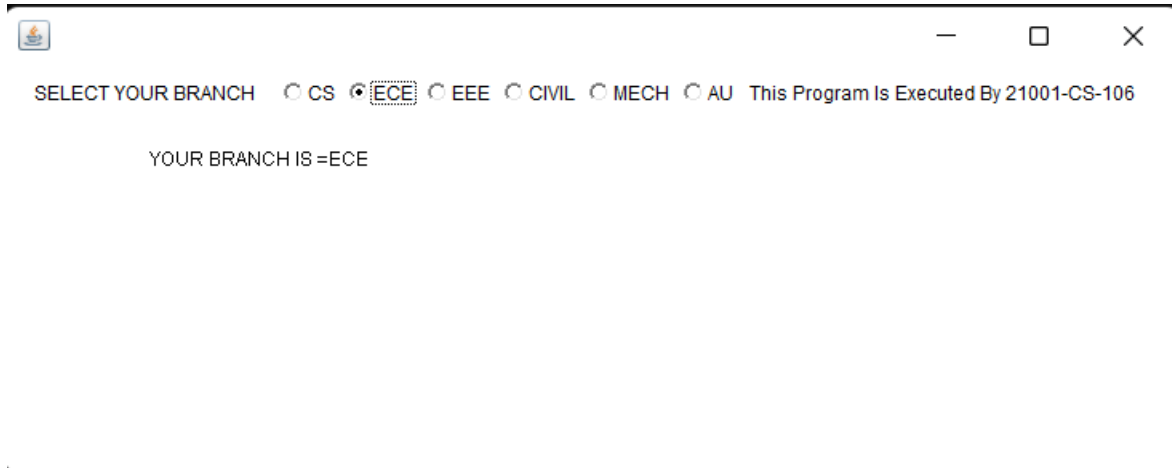


Q5. Write a java program to handle ItemStateChangedEvent generated by Radio Buttons

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
class RadioButtonEvent extends Frame implements ItemListener
{
    Label l1;
    String k;
    Checkbox c1,c2,c3,c4,c5,c6;
    Label name=new Label("This Program Is Executed By 21001-CS-106");
    public RadioButtonEvent()
    {
        setLayout(new FlowLayout());
        l1=new Label("SELECT YOUR BRANCH");
        CheckboxGroup cb=new CheckboxGroup();
        c1=new Checkbox("CS",cb,false);
        c2=new Checkbox("ECE",cb,false);
        c3=new Checkbox("EEE",cb,false);
        c4=new Checkbox("CIVIL",cb,false);
        c5=new Checkbox("MECH",cb,false);
        c6=new Checkbox("AU",cb,false);
        c1.addItemListener(this);
        c2.addItemListener(this);
        c3.addItemListener(this);
        c4.addItemListener(this);
        c5.addItemListener(this);
        c6.addItemListener(this);
        add(l1);
        add(c1);
        add(c2);
        add(c3);
        add(c4);
        add(c5);
        add(c6);
        add(name);
        setSize(300,300);
        setVisible(true);
    }
    public void itemStateChanged(ItemEvent e)
    {
        if(c1.getState()==true)
            k=k+c1.getLabel();
        if(c2.getState()==true)
            k=k+c2.getLabel();
        if(c3.getState()==true)
            k=k+c3.getLabel();
        if(c4.getState()==true)
            k=k+c4.getLabel();
        if(c5.getState()==true)
            k=k+c5.getLabel();
        if(c6.getState()==true)
            k=k+c6.getLabel();
        repaint();
    }
    public void paint(Graphics g)
```

```
{
    g.drawString("YOUR BRANCH IS =" + k, 100, 100);
    k = "";
}
public static void main(String s[])
{
    new RadioButtonEvent();
}
}
```

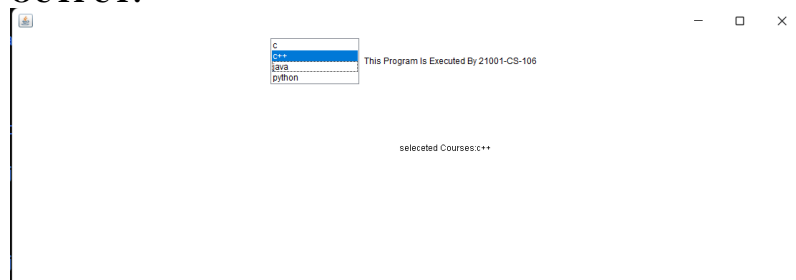
OUTPUT:

Q6. Write a java program to handle ActionEvent generated by Listbox

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
class ListEvent extends Frame implements ActionListener
{
    List Obj;
    FlowLayout Obj1;
    String[] S= new String[10];
    String f;
    Label name=new Label("This Program Is Executed By 21001-CS-106");
    public ListEvent()
    {
        Obj=new List(4,true);
        Obj1=new FlowLayout();
        setLayout(Obj1);
        Obj.add("c");
        Obj.add("c++");
        Obj.add("java");
        Obj.add("python");
        add(Obj);
        add(name);
        Obj.addActionListener(this);
        setVisible(true);
        setSize(300,300);
    }
    public void actionPerformed(ActionEvent e)
    {
        S=Obj.getSelectedItems();
        for(String a:S)
        {
            f=f+a+" ";
        }
        repaint();
    }
    public void paint(Graphics g)
    {
        g.drawString("seleceted Courses:"+f,550,200);
        f="";
    }
    public static void main(String[] args)
    {
        new ListEvent();
    }
}
```

OUTPUT:

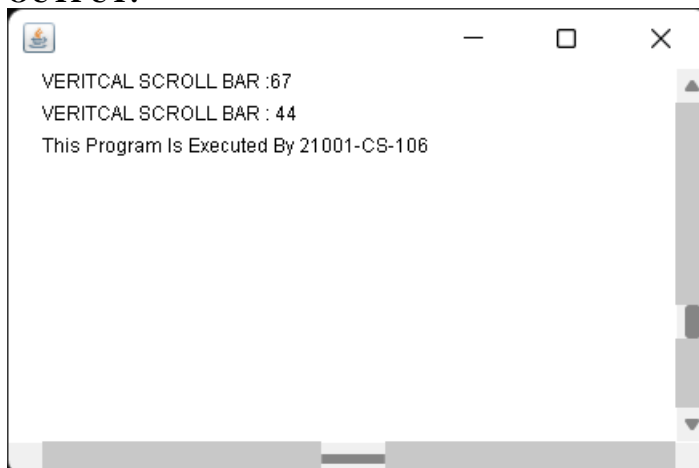


Q7. Write a java program with vertical and horizontal scroll bar.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
public class ExScrollBar extends Frame implements AdjustmentListener
{
    Scrollbar s1,s2;
    Label name=new Label("This Program Is Executed By 21001-CS-106");
    public ExScrollBar()
    {
        s1=new Scrollbar(Scrollbar.VERTICAL);
        s2=new Scrollbar(Scrollbar.HORIZONTAL);
        add(s1, BorderLayout.EAST);
        add(s2, BorderLayout.SOUTH);
        s1.addAdjustmentListener(this);
        s2.addAdjustmentListener(this);
        add(name);
        setSize(300,300);
        setVisible(true);
    }
    public void adjustmentValueChanged(AdjustmentEvent e)
    {
        repaint();
    }
    public void paint(Graphics g)
    {
        String s3="VERITCAL SCROLL BAR :"+s1.getValue();
        g.drawString(s3,30,50);
        s3="VERITCAL SCROLL BAR : "+s2.getValue();
        g.drawString(s3,30,70);
    }
    public static void main(String args[])
    {
        new ExScrollBar();
    }
}
```

OUTPUT:



Q8. Write a java program to handle mouse events.

PROGRAM :

```
import java.awt.*;
import java.awt.event.*;
public class ExMouse extends Frame implements MouseListener
{
    public ExMouse()
    {
        setLayout(new GridBagLayout());
        addMouseListener(this);
        setSize(300,300);
        setVisible(true);
    }
    public void mouseClicked(MouseEvent e)
    {
        System.out.println("mouseClicked at position "+e.getX()+" , "+e.getY());
    }
    public void mouseReleased(MouseEvent e)
    {
        System.out.println("mouseReleased at position "+e.getX()+" , "+e.getY());
    }
    public void mousePressed(MouseEvent e)
    {
        System.out.println("mousepressed at position "+e.getX()+" , "+e.getY());
    }
    public void mouseExited(MouseEvent e)
    {
        System.out.println("mouseExited at position "+e.getX()+" , "+e.getY());
    }
    public void mouseEntered(MouseEvent e)
    {
        System.out.println("mouseEntered at position "+e.getX()+" , "+e.getY());
    }
    public static void main(String args[])
    {
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-106");
        ExMouse e=new ExMouse();
    }
}
```

OUTPUT:

```
A:\Java Records>javac ExMouse.java

A:\Java Records>java ExMouse
THIS PROGRAM WAS EXECUTED BY 21001-CS-106
mouseEntered at position 286 , 278
mouseExited at position 293 , 228
mouseEntered at position 284 , 218
mousepressed at position 243 , 185
mouseReleased at position 243 , 185
mouseClicked at position 243 , 185
mouseExited at position 271 , 35
```

Q9. Write a java program to keyboard mouse events.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
public class ExKeyBoard extends Frame implements KeyListener
{
    public ExKeyBoard()
    {
        setLayout(new GridBagLayout());
        addKeyListener(this);
        setSize(300,300);
        setVisible(true);
    }
    public void keyReleased(KeyEvent e)
    {
        System.out.println("KEYRELEASED = "+e.getKeyChar());
    }
    public void keyPressed(KeyEvent e)
    {
        System.out.println("KEYPRESSED = "+e.getKeyChar());
    }
    public void keyTyped(KeyEvent e)
    {
        System.out.println("KEYTYPED = "+e.getKeyChar());
    }
    public static void main(String args[])
    {
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-106");
        ExKeyBoard e=new ExKeyBoard();
    }
}
```

OUTPUT:

```
A:\Java Records>javac ExKeyBoard.java

A:\Java Records>java ExKeyBoard
THIS PROGRAM WAS EXECUTED BY 21001-CS-106
KEYPRESSED = p
KEYTYPED = p
KEYRELEASED = p
KEYPRESSED = s
KEYTYPED = s
KEYRELEASED = s
```

Q10. Write a java program to window events.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
public class ExWindow extends Frame implements WindowListener
{
    public ExWindow()
    {
        setLayout(new GridBagLayout());
        addWindowListener(this);
        setSize(300,300);
        setVisible(true);
    }
    public void windowOpened(WindowEvent e)
    {
        System.out.println("WINDOW OPENED");
    }
    public void windowClosed(WindowEvent e)
    {
        System.out.println("WINDOW CLOSED");
    }
    public void windowClosing(WindowEvent e)
    {
        System.out.println("WINDOW CLOSING");
    }
    public void windowDeiconified(WindowEvent e)
    {
        System.out.println("WINDOW DEICONIFIED");
    }
    public void windowIconified(WindowEvent e)
    {
        System.out.println("WINDOW ICONIFIED");
    }
    public void windowActivated(WindowEvent e)
    {
        System.out.println("WINDOW ACTIVATED");
    }
    public void windowDeactivated(WindowEvent e)
    {
        System.out.println("WINDOW DEACTIVATED");
    }
    public static void main(String args[])
    {
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-106");
        ExWindow e=new ExWindow();
    }
}
```

OUTPUT:

```
A:\Java Records>javac ExWindow.java

A:\Java Records>java ExWindow
THIS PROGRAM WAS EXECUTED BY 21001-CS-106
WINDOW ACTIVATED
WINDOW OPENED
WINDOW CLOSING
WINDOW DEACTIVATED
```

Q11. Write a java program to create a frame with a menubar.

PROGRAM:

```
import java.awt.*;
import java.awt.event.*;
public class ExMenu extends Frame
{
    Menu m=new Menu("FILE");
    Label name=new Label("This Program Is Executed By 21001-CS-106");
    public ExMenu()
    {
        MenuBar mb=new MenuBar();
        setMenuBar(mb);
        mb.add(m);
        m.add("NEW");
        m.add("OPEN");
        m.add("SAVE ");
        m.add("SAVE AS");
        m.add("EXIT");
        add(name);
        setSize(300,300);
        setVisible(true);
    }
    public static void main(String args[])
    {
        ExMenu e=new ExMenu();
    }
}
```

OUTPUT:

