GCD-APPLET

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code=gcd_applet width=500 height=200></applet>*/
public class gcd_appplet
   Label l1,l2,l3;
   TextField t1,t2,t3;
   Button b1,b2;
   public void init(){

      l1 = new Label("Num1 ");
      l2 = new Label("Num2 ");
      l3 = new Label("Sum ");
      t1 = new TextField();
      t2 = new TextField();
      t3 = new TextField();
      b1 = new Button("Submit");
      b2 = new Button("Clear");

      setLayout(new GridLayout(4,2));
      add(l1);
      add(t1);
      add(l2);
      add(t2);
      add(l3);
      add(t3);
      add(b1);
      add(b2);
      t3.setEditable(false);
      b1.addActionListener(this);
      b2.addActionListener(this);
   }
   public void actionPerformed(ActionEvent ae){
      int a=Integer.parseInt(t1.getText());
      int b=Integer.parseInt(t2.getText());
      if(ae.getSource()==b1){

         while(a!=b){
            if(a>b)
               a-=b;
            else
               b-=a;
         }
         t3.setText("GCD: "+a);
      }
      if(ae.getSource()==b2){
         t1.setText("");
         t2.setText("");
         t3.setText("");
         t1.requestFocus();
```

```
    }
  }
}
```

COMPILE: javac gcd_applet.java

RUN: appletviewer gcd_applet

GCD-SWING

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class gcd_swing extends JApplet implements ActionListener{
  JLabel l1,l2,l3;
  JTextField t1,t2,t3;
  JButton b1,b2;
  public void init(){
    Container cp = getContentPane();
    cp.setLayout(new GridLayout(4,2));
    l1 = new JLabel("Num1 ");
    l2 = new JLabel("Num2 ");
    l3 = new JLabel("Sum ");
    t1 = new JTextField(10);
    t2 = new JTextField(10);
    t3 = new JTextField(10);
    b1 = new JButton("Submit");
    b2 = new JButton("Clear");

    add(l1);
    add(t1);
    add(l2);
    add(t2);
    add(l3);
    add(t3);
    add(b1);
    add(b2);

    b1.addActionListener(this);
    b2.addActionListener(this);
  }
  public void actionPerformed(ActionEvent ae){
    int a=Integer.parseInt(t1.getText());
    int b=Integer.parseInt(t2.getText());
    if(ae.getSource()==b1){

      while(a!=b){
        if(a>b)
          a-=b;
        else
```

```
            b-=a;
          }
          t3.setText("GCD: "+a);
        }
        if(ae.getSource()==b2){
          t1.setText("");
          t2.setText("");
          t3.setText("");
          t1.requestFocus();
        }
    }
    public static void main(String[] args){
        JFrame frame = new JFrame(" JFrame");
        gcd_swing gcd = new gcd_swing();
        frame.add(gcd);
        frame.setSize(800,400);
        frame.setDefaultCloseOperation(frame.EXIT_ON_CLOSE);
        gcd.init();
        frame.setVisible(true);

    }
}
```

COMPILE: javac gcd_swing.java

RUN: java gcd_swing

EMP-APPLET

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code=emp width=400 height=400></applet>*/
public class emp extends Applet implements ActionListener{
    Label l1,l2,l3,l4;
    TextField t1,t2,t3,t4;
    Button b1,b2;
    public void init(){
        setBackground(Color.yellow);
        setForeground(Color.red);
        l1=new Label("Salary ");
        l2=new Label("DA ");
        l3=new Label("HRA ");
        l4=new Label("Total ");
        t1=new TextField();
        t2=new TextField();
        t3=new TextField();
        t4=new TextField();
        b1=new Button("submit");
        b2=new Button("clear");

        setLayout(new GridLayout(5,2));
```

```
      add(l1);
      add(t1);
      add(l2);
      add(t2);
      add(l3);
      add(t3);
      add(l4);
      add(t4);
      add(b1);
      add(b2);
      b1.addActionListener(this);
      b2.addActionListener(this);
   }
   public void actionPerformed(ActionEvent ae){
         float da,hra,ssal;

      if(ae.getSource()==b1){
         float a=Float.parseFloat(t1.getText());
         if(a<20000){
            da=a*0.5f;
            hra=a*0.2f;
         }else{
            da=a*0.9f;
            hra=a*0.5f;
         }

         ssal=a+da+hra;
         t2.setText("DA "+Float.toString(da));
         t3.setText("HRA "+Float.toString(hra));
         t4.setText("Salary: "+Float.toString(ssal));

      }
      if(ae.getSource()==b2){
         t1.setText("");
         t2.setText("");
         t3.setText("");
         t4.setText("");
         t1.requestFocus();
      }
   }
}
```

COMPILE : javac emp.java

RUN: appletviewer emp.java

EMP-SWING

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class emp_swing extends JApplet implements ActionListener{
   JLabel l1,l2,l3,l4;
   JTextField t1,t2,t3,t4;
   JButton b1,b2;
```

```
public void init(){
   Container cp= getContentPane();
   cp.setLayout(new GridLayout(5,2));

   l1 = new JLabel("salary ");
   l2 = new JLabel("DA ");
   l3 = new JLabel("HRA ");
   l4 = new JLabel("Sum ");
   t1 = new JTextField(10);
   t2 = new JTextField(10);
   t3 = new JTextField(10);
   t4 = new JTextField(10);
   b1 = new JButton("Submit");
   b2 = new JButton("Clear");


   add(l1);
   add(t1);
   add(l2);
   add(t2);
   add(l2);
   add(t2);
   add(l3);
   add(t3);
   add(l4);
   add(t4);
   add(b1);
   add(b2);
   add(b2);



   b1.addActionListener(this);
   b2.addActionListener(this);
}
public void actionPerformed(ActionEvent ae){
   if(ae.getSource()==b1){
   float sal,da=0,hra=0,sum;
   int a=Integer.parseInt(t1.getText());

      if(a<15000){
         da=0.5f*a;
         hra=0.2f*a;
      }else{
         da=0.9f*a;
         hra=0.5f*a;
      }
      t2.setText("DA: "+da);
      t3.setText("HRA: "+hra);
      sum=a+da+hra;

     t4.setText(" "+sum);

   }
   if(ae.getSource()==b2){
```

```
        t1.setText("");
        t2.setText("");
        t3.setText("");
        t4.setText("");
        t1.requestFocus();
      }
   }
    public static void main(String[] args){
      JFrame frame = new JFrame("Frame Swings");
      emp_swing emp=new emp_swing();
      frame.add(emp);
      frame.setSize(400,400);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      emp.init();
      frame.setVisible(true);
   }
}
```

COMPILE : javac emp_swing.java

RUN: java emp_swing.java

FIBBONOCCI-APPLET

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code= fib_applet width=400 height=400></applet>*/
public class fib_applet extends Applet implements ActionListener{
   Label l1,l2,l3;
   TextArea ta1;
   TextField t1,t2;
   Button b1,b2;
   public void init(){
      setLayout(new GridLayout(4,2));
      l1 = new Label("Num:");
      add(l1);
      t1 = new TextField(10);
      add(t1);
      l2 = new Label("Series:");
      add(l2);
      ta1 = new TextArea(1,10);
      add(ta1);
      l3 = new Label("Sum");
      add(l3);
      t2 = new TextField(10);
      add(t2);
      b1 = new Button("Submit");
      b1.addActionListener(this);
      add(b1);
      b2 = new Button("Clear");
      b2.addActionListener(this);
      add(b2);

   }
   public void actionPerformed(ActionEvent ae){
      int fib=0,sum=0,n1=0,n2=1;
```

```
      if(ae.getSource()==b1){
      int n=Integer.parseInt(t1.getText());
      ta1.setText(" ");
      ta1.append(n1+" "+n2);
      for(int i=0;i<n;i++){
         fib=n1+n2;
         sum+=fib;
         n1=n2;
         n2=fib;
         ta1.append(" "+fib);
      }
      t2.setText(""+sum);
   }
   if(ae.getSource()==b2){
      t1.setText("");
      t2.setText("");
      ta1.setText("");
      t1.requestFocus();
   }
   }

}
```

COMPILE javac fib_applet.java

RUN appletviewer fib_applet.java

FIBONOCCI- SWING

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class fib_applet extends JApplet implements ActionListener{
   JLabel l1,l2,l3;
   JTextArea ta1;
   JTextField t1,t2;
   JButton b1,b2;
   public void init(){
      Container cp = getContentPane();
      cp.setLayout(new GridLayout(4,2));
      l1 = new JLabel("Num:");
      add(l1);
      t1 = new JTextField(10);
      add(t1);
      l2 = new JLabel("Series:");
      add(l2);
      ta1 = new JTextArea(1,10);
      add(ta1);
      l3 = new JLabel("Sum");
      add(l3);
      t2 = new JTextField(10);
      add(t2);
      b1 = new JButton("Submit");
      b1.addActionListener(this);
      add(b1);
      b2 = new JButton("Clear");
      b2.addActionListener(this);
```

```
      add(b2);

   }
   public void actionPerformed(ActionEvent ae){
      int fib=0,sum=0,n1=0,n2=1;
      if(ae.getSource()==b1){
      int n=Integer.parseInt(t1.getText());
      ta1.setText(" ");
      ta1.append(n1+" "+n2);
      for(int i=0;i<n;i++){
         fib=n1+n2;
         sum+=fib;
         n1=n2;
         n2=fib;
         ta1.append(" "+fib);
      }
      t2.setText(""+sum);
   }
   if(ae.getSource()==b2){
      t1.setText("");
      t2.setText("");
      ta1.setText("");
      t1.requestFocus();
   }
   }
   public static void main(String[] args){
      JFrame frame = new JFrame("SWING");
      fib_applet fg = new fib_applet();
      frame.add(fg);
      frame.setSize(800,300);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      fg.init();
      frame.setVisible(true);
   }
}
```

COMPILE javac fib_swing.java

RUN: java fib_swing

Write a java program to demonstrate up-casting and compile-time polyphism.

```
import java.util.Scanner;
import java.io.*;
class Shape
{
   public void area(int a,int b)
   {
      System.out.println("Shape Area");
   }
   public void area(int a,int b,int c)
   {
      System.out.println("Shape Area");
   }
```

```
}
class Box extends Shape
{
   public void area(int a,int b)
   {
      System.out.println("Area of rectangle: "+a*b);
   }
   public void area(int a,int b,int c)
   {
      System.out.println("Area of Box: "+a*b*c);
   }
}
public class upcast_polymorphism
{
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
      Shape s = new Box();
      Box bx = new Box();
      int a,b,c;
      a=5;
      b=3;
      c=2;
      while(true)
      {
         System.out.println("Area of \n1. Rectangle\n2. Box\nChoice: ");
         int ch=sc.nextInt();
         switch(ch)
         {
         case 1:
            s.area(a,b);
            bx.area(a,b);
            break;
         case 2:
            s.area(a,b,c);
            bx.area(a,b,c);
            break;
         default:
            System.exit(0);
         }
      }

   }
}
```