**1.FACTORIAL USING RECURSION**

import java.io.\*;

class factorial

{

int fact(int n)

{

int result;

if(n==1)

result=1;

else

result=n\*fact(n-1);

return result;

}

}

class fact

{

public static void main(String args[])

{

factorial f=new factorial();

String j;

int n=0;

System.out.println("enter a number");

DataInputStream dn= new DataInputStream(System.in);

try

{

j=dn.readLine();

n=Integer.parseInt(j);

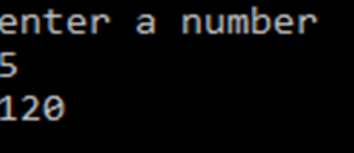
}catch(Exception e){}

System.out.println(f.fact(n));

}

}

**OUTPUT**

****

**2.METHOD OVERLOADING**

import java.io.\*;

class method

{

int i=0;

void power(int a,int b)

{

int p=1;

for(i=1;i<=b;i++)

{

p=p\*a;

}

System.out.println("power is "+p);

}

void power(double a,int b)

{

double p=1;

for(i=1;i<=b;i++)

{

p=p\*a;

}

System.out.println("power is"+p);

}

}

class poverload

{

public static void main(String args[])

{

method m= new method();

m.power(2,3);

m.power(4.5,2);

}

}

**OUTPUT**



**3.DYNAMIC METHOD DISPATCH**

import java.io.\*;

class A

{

void method()

{

System.out.println("inside A");

}

}

class B extends A

{

void method()

{

System.out.println("inside B");

}

}

class C extends A

{

void method()

{

System.out.println("inside C");

}

}

class Dispatch

{

public static void main(String args[])

{

A a=new A();

B b=new B();

C c=new C();

A r;

r=a;

r.method();

r=b;

r.method();

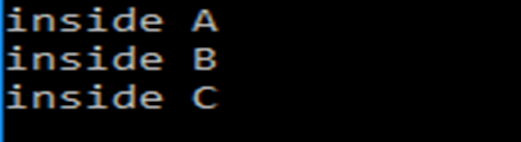
r=c;

r.method();

}

}

**OUTPUT**

****

**4.ODD OR EVEN**

import java.io.\*;

class check

{

void checkdata(int n)

{

if(n%2==0)

{

System.out.println("even number");

}

else

{

System.out.println("odd number");

}

}

}

class oddoreven{

public static void main(String []args)

{

int j=0;

check c=new check();

System.out.println("enter a number");

DataInputStream n= new DataInputStream(System.in);

try

{

j=n.readInt();

}catch(Exception e){

}

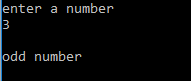
c.checkdata(j);

}

}

**OUTPUT**





**5.FIBONACCI**

import java.io.\*;

class Fib

{

int n1=0,n2=1,sum=0,i=1;

void find(int n)

{

do

{

sum=n1+n2;

System.out.println(sum);

n1=n2;

n2=sum;

i++;

}while(i<=n);

}

}

class Fibmain

{

public static void main(String args[])

{

String j;

int n=0;

Fib f=new Fib();

System.out.println("enter a number");

DataInputStream dn= new DataInputStream(System.in);

try

{

j=dn.readLine();

n=Integer.parseInt(j);

System.out.println(n);

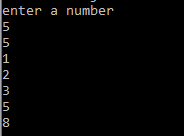
}catch(Exception e){}

f.find(n);

}

}

**OUTPUT**

****

**6.PRIME OR NOT**

import java.io.\*;

class prime

{

int i,flag=0;

void primecheck(int n)

{

for(i=2;i<=n/2;i++)

{

if(n%i==0)

{

System.out.println("not prime ");

flag=1;

}

}

if(flag==0)

{

System.out.println("prime");

}

}

}

class checkprime

{

public static void main(String args[])

{

prime p=new prime();

String j;

int n=0;

System.out.println("enter a number");

DataInputStream dn= new DataInputStream(System.in);

try

{

j=dn.readLine();

n=Integer.parseInt(j);

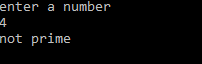
}catch(Exception e){}

p.primecheck(n);

}

}

**OUTPUT**

**7.INTERFACE**

class student

{

int rollnumber;

void getnumber(int n){

rollnumber=n;

}

void putnumber()

{

System.out.println("rollnumber:"+rollnumber);

}

}

class test extends student

{

int p1,p2;

void getmark(int m1,int m2)

{

p1=m1;

p2=m2;

}

void putmark()

{

System.out.println("marks"+p1);

System.out.println("marks"+p2);

}

}

interface sports

{

int sptwt=10;

void putwt();

}

class result extends test implements sports

{

int total;

public void putwt()

{

System.out.println("sports weight"+sptwt);

}

void display()

{

putnumber();

putmark();

putwt();

total=p1+p2+sptwt;

System.out.println("total mark "+total);

}

}

class hybrid

{

public static void main(String args[])

{

result std=new result();

std.getnumber(1234);

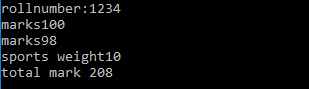
std.getmark(100,98);

std.display();

}

}

**OUTPUT**



**8.STRING**

class count

{

public static void main(String args[])

{

String a="assumption";

int i,c=0;

for(i=0;i<a.length();i++)

{

if(a.charAt(i)=='s')

c=c+1;

}

System.out.println("no:of s is"+c);

}

}

**OUTPUT**



**9.PROMOTION**

import java.io.\*;

class promotion

{

public static void main(String args[])

{

byte b=42;

char c='a';

short s=1024;

int i=50000;

float f=5.67f;

double d=.1234;

double result=(f\*b)+(i/c)-(d\*s);

System.out.println(+(f\*b)+"+"+(i/c)+"-"+(d\*s));

System.out.println("result ="+result);

}

}

**OUTPUT**



**10.TYPE CASTING**

class conversion

{

public static void main(String args[])

{

byte b;

int t=257;

double d=323.142;

System.out.println("conversion of int to byte");

b=(byte)t;

System.out.println("converted "+t+" "+b);

System.out.println("conversion of double into int");

t=(int)d;

System.out.println("converted "+d+" "+t);

System.out.println("conversion of double into byte");

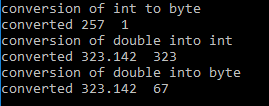
b=(byte)d;

System.out.println("converted "+d+" "+b);

}

}

**OUTPUT**



**11.REVERSE OF A NUMBER**

import java.io.\*;

class Reverse

{

int digit,rev=0;

void check(int n)

{

do

{

digit=n%10;

rev=rev\*10+digit;

n=n/10;

}while(n!=0);

System.out.println("reverse of the number "+rev);

}

}

class Remain{

public static void main(String args[])

{

int n=0;

String j;

Reverse r=new Reverse();

System.out.println("enter a number");

DataInputStream dn= new DataInputStream(System.in);

try

{

j=dn.readLine();

n=Integer.parseInt(j);

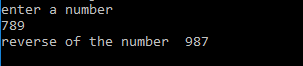
}catch(Exception e){}

r.check(n);

}

}

**OUTPUT**



**12. STACK IMPLEMENTATION-PACKAGE**

package package1;

import java.io.\*;

public class stack

{

int stk[]=new int[10];

int top;

public stack()

{

top=-1;

}

public void push(int item)

{

top++;

stk[top]=item;

System.out.println("inserted"+item);

if(top>=9)

{

System.out.println("overflow");

}

}

public void pop()

{

if(top==-1)

{

System.out.println("empty");

}

else

{

System.out.println("deleted "+stk[top]);

top--;

}

}

}

import java.io.\*;

import package1.stack;

class teststack

{

public static void main(String args[])

{

stack s=new stack();

for(int i=0;i<10;i++)

{

s.push(i);

}

for(int i=-1;i<10;i++)

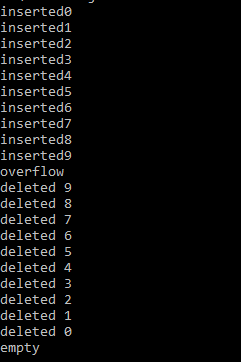
{

s.pop();

}

}

}

****

**13.THREAD (IMPLEMENTS RUNNABLE)**

class NewThread implements Runnable

{

Thread t;

NewThread()

{

t= new Thread(this,"demo thread");

System.out.println("child thread:"+ t);

t.start();

}

public void run()

{

try

{

for(int i=5;i>0;i--)

{

System.out.println("child thread"+i);

Thread.sleep(500);

}

}

catch(InterruptedException e)

{

System.out.println("child interrupted");

}

System.out.println("exiting new thread");

}

}

class ThreadDemo

{

public static void main(String args[])

{

new NewThread();

try

{

for(int i=5;i>0;i--)

{

System.out.println("main thread:"+i);

Thread.sleep(1000);

}

}catch(InterruptedException e)

{

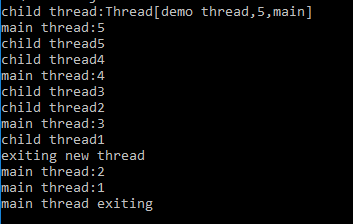
System.out.println("main thread interrupted");

}

System.out.println("main thread exiting");

}}

**OUTPUT**



**14.THREAD (EXTENDS THREAD)**

class newThread extends Thread

{

newThread()

{

super("demothread");

System.out.println("childthread"+this);

start();

}

public void run()

{

try

{

for(int i=5;i>0;i--)

{

System.out.println("child thread"+i);

Thread.sleep(500);

}

}

catch(InterruptedException e)

{

System.out.println("child interrupted");

}

System.out.println("exiting child thread");

}

}

class ExtendThread

{

public static void main(String args[])

{

new newThread();

try

{

for(int i=5;i>0;i--)

{

System.out.println("main thread" +i);

Thread.sleep(1000);

}

}

catch(InterruptedException e)

{

System.out.println("main thread interrupted");

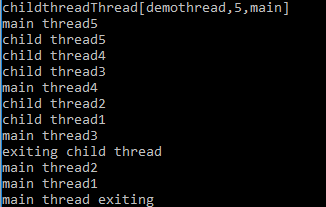
}

System.out.println("main thread exiting");

}

}

**OUTPUT**



**15.ACCESS CONTROL PACKAGE INHERITANCE**

**Package1**

Demo.java

package p1;

class demo

{

public static void main(String args[])

{

Protection o1=new Protection();

Derived o2=new Derived();

SamePackage o3=new SamePackage();

}

}

Derived.java

package p1;

class Derived extends p1.Protection

{

Derived()

{

System.out.println("derived constructor");

System.out.println("n = " + n);

System.out.println("n\_pro = " + n\_pro);

//System.out.println("n\_pri = " + n\_pri);//not possible to print private data

System.out.println("n\_pub = " + n\_pub);

}

}

Samepackage.java

package p1;

class SamePackage

{

SamePackage()

{

Protection p=new Protection();

System.out.println("same package constructor");

System.out.println("n = " + p.n);

//System.out.println("n\_pri = " + p.n\_pri);

System.out.println("n\_pro = " + p.n\_pro);

System.out.println("n\_pub = " + p.n\_pub);

}}

Protection.java

package p1;

public class Protection

{

int n=1;

private int n\_pri=2;

protected int n\_pro=3;

public int n\_pub=4;

public Protection()

{

System.out.println("base constructor");

System.out.println("n = " + n);

System.out.println("n\_pri = " + n\_pri);

System.out.println("n\_pro = " + n\_pro);

System.out.println("n\_pub = " + n\_pub);

}

}

**Package 2**

Demo.java

package p2;

public class demo

{

public static void main(String args[])

{

Protection2 oo1=new Protection2();

OtherPackage oo2=new OtherPackage();

}

}

Otherpackage.java

package p2;

class OtherPackage {

OtherPackage() {

p1.Protection p = new p1.Protection();

System.out.println("other package constructor");

// System.out.println("n = " + p.n);

// System.out.println("n\_pri = " + p.n\_pri);

// System.out.println("n\_pro = " + p.n\_pro);

System.out.println("n\_pub = " + p.n\_pub);

}

}

Protection2.java

package p2;

public class Protection2 extends p1.Protection {

Protection2() {

System.out.println("derived other package constructor");

// System.out.println("n = " + n);

// System.out.println("n\_pri = " + n\_pri);

System.out.println("n\_pro = " + n\_pro);

System.out.println("n\_pub = " + n\_pub);

}

}

**OUTPUT**

**16.EXCEPTION**

import java.io.\*;

class tryclass

{

public static void main(String args[])

{

int a[]={5,10};

int b=5;

int j=0;

String s;

System.out.println("options select 1 for arithmetic exception 2 for arrayindex");

DataInputStream o= new DataInputStream(System.in);

try{

s=o.readLine();

j=Integer.parseInt(s);

}catch(Exception e){}

switch(j)

{

case 1:

try

{

int x=a[1]/0;

}

catch(ArithmeticException e)

{

System.out.println("division by zero"+e);

}

break;

case 2:

try

{

int x=a[2]/0;

}

catch(ArrayIndexOutOfBoundsException e)

{

System.out.println("array out of boundary"+e);

}

break;

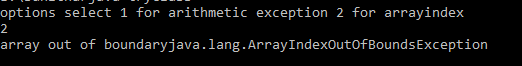
}

}

}

**OUTPUT**





**17.FINALLY**

class FinallyDemo

{

static void methodA()

{

try

{

System.out.println("inside A");

throw new RuntimeException("demo");

}

finally

{

System.out.println("methods A's finally");

}

}

static void methodB()

{

try

{

System.out.println("inside methodB");

return;

}

finally

{

System.out.println("methods B's finally");

}

}

static void methodC()

{

try

{

System.out.println("inside method c");

}

finally

{

System.out.println("method c's finally");

}

}

public static void main(String args[])

{

try

{

methodA();

}

catch(Exception e)

{

System.out.println("exception catched");

}

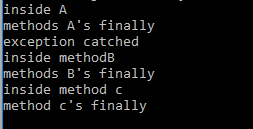
methodB();

methodC();

}

}

**OUTPUT**



**18.CUSTOMIZED EXCEPTION**

import java.io.\*;

class newException extends Exception

{

int a;

newException(int b)

{

a=b;

}

public String toString()

{

return "newException("+a+")";

}

}

class DemoException

{

static void calculate(int a,int b) throws newException

{

int sum=a+b;

System.out.println(sum);

if(sum==0)

{

System.out.println("one is positive number and other is same negative number");

throw new newException(a);

}

else

{

System.out.println("normal calculation");

System.out.println("normal exit");

}

}

public static void main(String args[])

{

try

{

calculate(30,30);

calculate(-30,30);

}

catch(newException e)

{

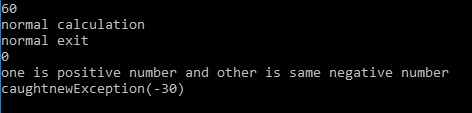
System.out.println("caught"+e);

}

}

}

**OUTPUT**



**19.REGISTRATION FORM**

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

class Application implements ActionListener

{

Frame f;

TextField name,address,emailid,dob,qualification;

Button B1,B2;

Application()

{

f=new Frame();

f.setVisible(true);

f.setSize(1000,1000);

Label l1=new Label("name");

f.add(l1);

l1.setBounds(100,100,70,20);

Label l2=new Label("address");

f.add(l2);

l2.setBounds(100,200,70,20);

Label l3=new Label("emailid");

f.add(l3);

l3.setBounds(100,300,70,20);

Label l4=new Label("dob");

f.add(l4);

l4.setBounds(100,400,70,20);

Label l5=new Label("qualification");

f.add(l5);

l5.setBounds(100,500,50,10);

name=new TextField(20);

f.add(name);

name.setBounds(400,100,70,20);

address=new TextField(20);

f.add(address);

address.setBounds(400,200,70,20);

emailid=new TextField(20);

f.add(emailid);

emailid.setBounds(400,300,70,20);

dob=new TextField(20);

f.add(dob);

dob.setBounds(400,400,70,20);

qualification=new TextField(20);

f.add(qualification);

qualification.setBounds(400,500,70,20);

B1=new Button("SUBMIT");

f.add(B1);

B1.setBounds(400,600,100,30);

B2=new Button("CANCEL");

f.add(B2);

B2.setBounds(500,600,100,30);

B1.addActionListener(this);

B2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getActionCommand()=="CANCEL")

{

f.dispose();

System.exit(0);

}

}

}

class Amain

{

public static void main(String args[])

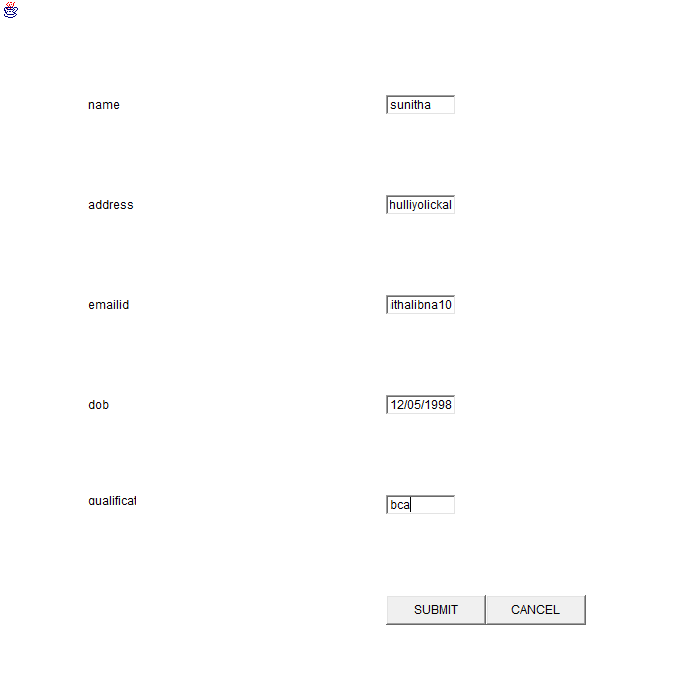
{

Application a=new Application();

}

}

**OUTPUT**



**20.EVENT HANDLING**

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

class events implements MouseListener

{

Button b;

Frame f;

Label l;

events()

{

f=new Frame();

f.setVisible(true);

l=new Label(" mouse ");

f.add(l);

l.setBounds(100,100,100,50);

b=new Button("button");

f.add(b);

b.setBounds(100,100,70,20);

b.addMouseListener(this);

}

public void mouseClicked(MouseEvent me)

{

l.setText("mouse clicked");

}

public void mouseEntered(MouseEvent me)

{

l.setText("mouse entered");

}

public void mousePressed(MouseEvent me)

{

l.setText("mouse pressed");

}

public void mouseExited(MouseEvent me)

{

l.setText("mouse exited");

}

public void mouseReleased(MouseEvent me)

{

l.setText("mouse released");

}

}

class Eventscl

{

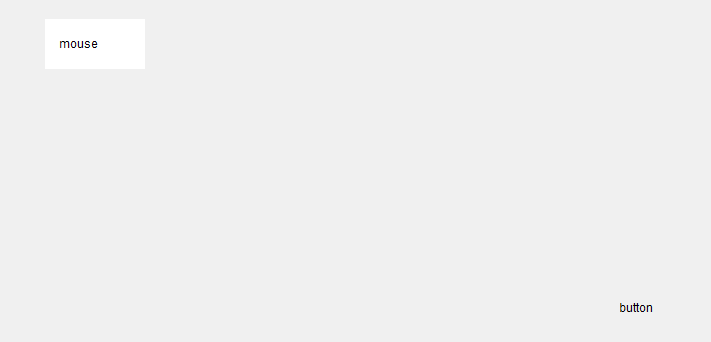
public static void main(String args[])

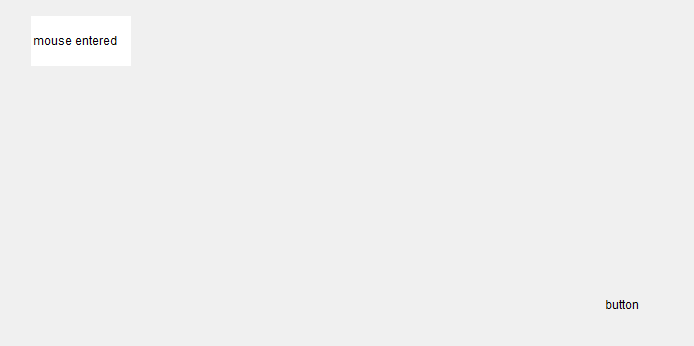
{

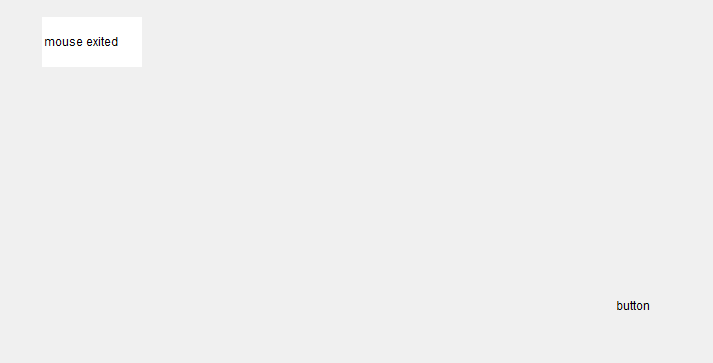
events e=new events();

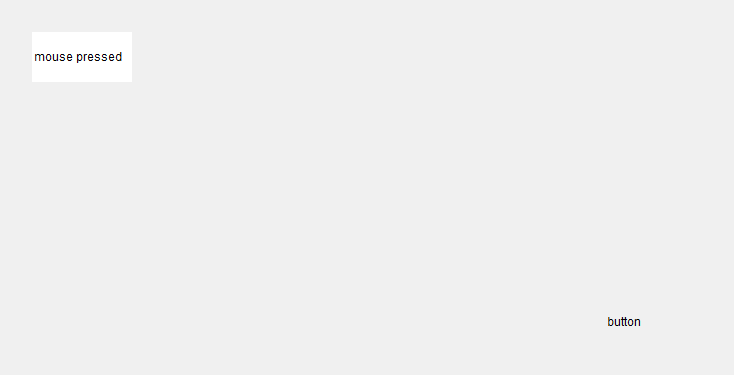
}

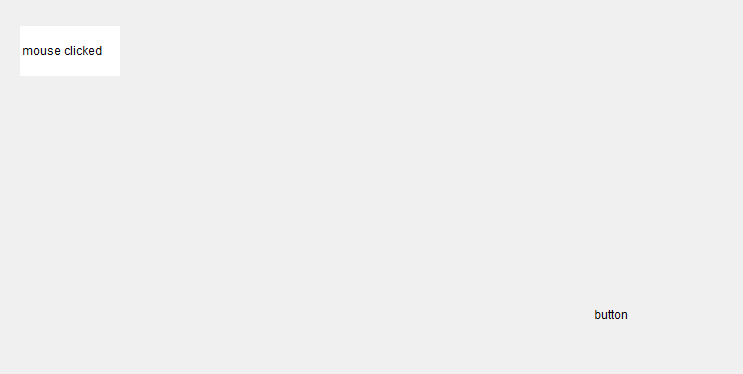
}

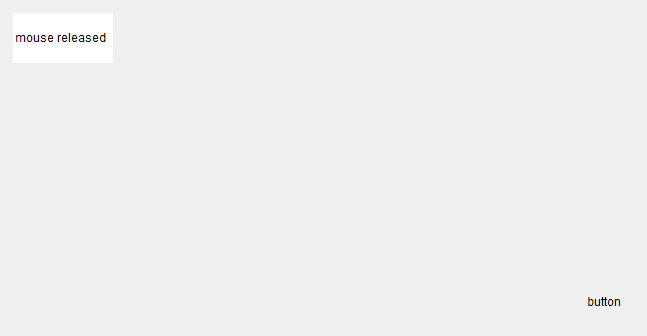












**21.SIMPLE BANNER APPLET**

import java.awt.\*;

import java.applet.\*;

import java.lang.\*;

/\*

<applet code="simplebanner" width=300 height=50>

</applet>

\*/

public class simplebanner extends Applet implements Runnable

{

String msg="A MOVING BANNER ";

Thread t=null;

int state;

boolean stopFlag;

public void init()

{

setBackground(Color.cyan);

setForeground(Color.red);

}

public void start()

{

t=new Thread(this);

stopFlag=false;

t.start();

}public void run()

{

char ch;

for(;;)

{

try

{

repaint();

Thread.sleep(200);

ch=msg.charAt(0);

msg=msg.substring(1,msg.length());

msg+=ch;

if(stopFlag)

break;

}

catch(InterruptedException e){}

}

}

public void stop()

{

stopFlag=true;

t=null;

}

public void paint(Graphics g)

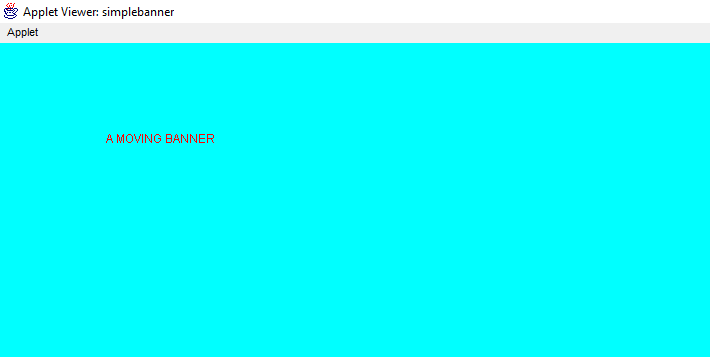
{

g.drawString(msg,100,50);

}

}

**OUTPUT**



**22.JDBC**

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.sql.\*;

class Application implements ActionListener

{

Frame f;

TextField name,address,emailid,dob,qualification;

Button B1,B2;

Application()

{

f=new Frame();

f.setVisible(true);

f.setSize(1000,1000);

Label l1=new Label("name");

f.add(l1);

l1.setBounds(100,100,70,20);

Label l2=new Label("address");

f.add(l2);

l2.setBounds(100,200,70,20);

Label l3=new Label("emailid");

f.add(l3);

l3.setBounds(100,300,70,20);

Label l4=new Label("dob");

f.add(l4);

l4.setBounds(100,400,70,20);

Label l5=new Label("qualification");

f.add(l5);

l5.setBounds(100,500,50,10);

name=new TextField(20);

f.add(name);

name.setBounds(400,100,70,20);

address=new TextField(20);

f.add(address);

address.setBounds(400,200,70,20);

emailid=new TextField(20);

f.add(emailid);

emailid.setBounds(400,300,70,20);

dob=new TextField(20);

f.add(dob);

dob.setBounds(400,400,70,20);

qualification=new TextField(20);

f.add(qualification);

qualification.setBounds(400,500,70,20);

B1=new Button("SUBMIT");

f.add(B1);

B1.setBounds(400,600,100,30);

B2=new Button("CANCEL");

f.add(B2);

B2.setBounds(500,600,100,30);

B1.addActionListener(this);

B2.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getActionCommand()=="CANCEL")

{

f.dispose();

System.exit(0);

}

if(e.getActionCommand()=="SUBMIT")

{

try

{

String s1,s2,s3,s4,s5;

s1=name.getText();

s2=address.getText();

s3=emailid.getText();

s4=dob.getText();

s5=qualification.getText();

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

Connection conn=DriverManager.getConnection("jdbc:odbc:ds");

String sql="INSERT INTO Table1 values(?,?,?,?,?)";

PreparedStatement s=conn.prepareStatement(sql);

s.setString(1,s1);

s.setString(2,s2);

s.setString(3,s3);

s.setString(4,s4);

s.setString(5,s5);

s.executeUpdate();

s.close();

conn.close();

}

catch(Exception e1)

{

System.out.println(e1);

}

System.out.println("INSERTED");

}

}

}

class Amain

{

public static void main(String args[])

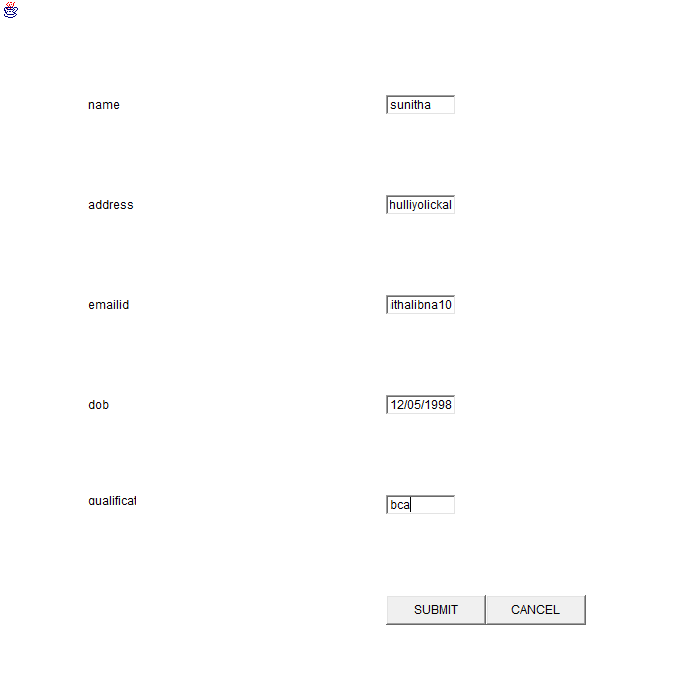
{

Application a=new Application();

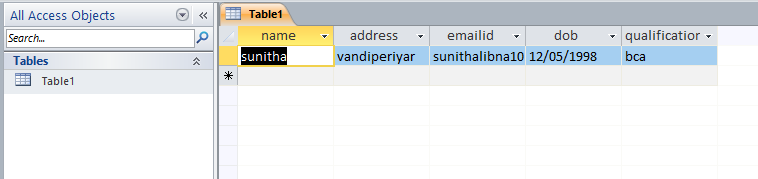
}

}

**OUTPUT**







**23.CALCULATOR**

import java.awt.\*;

import java.awt.event.\*;

public class Calcu implements ActionListener

{

int c,n;

Frame f=new Frame("My Calculator");

String s1,s2,s3,s4,s5,b;

Button b1=new Button("0");

Button b2=new Button("1");

Button b3=new Button("2");

Button b4=new Button("3");

Button b5=new Button("4");

Button b6=new Button("5");

Button b7=new Button("6");

Button b8=new Button("7");

Button b9=new Button("8");

Button b10=new Button("9");

Button b11=new Button("+");

Button b12=new Button("-");

Button b13=new Button("\*");

Button b14=new Button("/");

Button b15=new Button("%");

Button b16=new Button("=");

Button b17=new Button("c");

Panel p=new Panel();

GridLayout g=new GridLayout(4,4,10,20);

TextField tf= new TextField(20);

Calcu()

{

f.setLayout(new FlowLayout());

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

b6.addActionListener(this);

b7.addActionListener(this);

b8.addActionListener(this);

b9.addActionListener(this);

b10.addActionListener(this);

b11.addActionListener(this);

b12.addActionListener(this);

b13.addActionListener(this);

b14.addActionListener(this);

b15.addActionListener(this);

b16.addActionListener(this);

b17.addActionListener(this);

f.add(tf);

p.setLayout(g);

p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);

p.add(b6);p.add(b7);p.add(b8);p.add(b9);p.add(b10);

p.add(b11);p.add(b12);p.add(b13);p.add(b14);

p.add(b15);p.add(b16);p.add(b17);f.add(p);

f.setSize(new Dimension(400,400));

f.setVisible(true);

}

public void actionPerformed(ActionEvent e){

b=e.getActionCommand();

if(b.equals("0")||b.equals("1")||b.equals("2")||b.equals("3")||b.equals("4")||b.equals("5")||b.equals("6")||b.equals("7")||b.equals("8")||b.equals("9"))

{

s3=tf.getText()+b;

tf.setText(s3);

}

if(e.getSource()==b11)

{

s1=tf.getText();

tf.setText("");

c=1;

}

if(e.getSource()==b12)

{

s1=tf.getText();

tf.setText("");

c=2;

}

if(e.getSource()==b13)

{

s1=tf.getText();

tf.setText("");

c=3;

}

if(e.getSource()==b14)

{

s1=tf.getText();

tf.setText("");

c=4;

}

if(e.getSource()==b15)

{

s1=tf.getText();

tf.setText("");

c=5;

}

if(e.getSource()==b16)

{

s2=tf.getText();

if(c==1)

{

n=Integer.parseInt(s1)+Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

else

if(c==2)

{

n=Integer.parseInt(s1)-Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

else

if(c==3)

{

n=Integer.parseInt(s1)\*Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

if(c==4)

{

try

{

int p=Integer.parseInt(s2);

if(p!=0)

{

n=Integer.parseInt(s1)/Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

else

tf.setText("infinite");

}

catch (Exception i){}

}

if(c==5)

{

n=Integer.parseInt(s1)%Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

}

if(e.getSource()==b17)

{

tf.setText("");

}

}

public static void main (String args[])

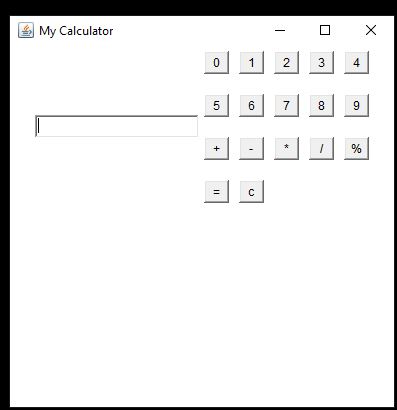
{

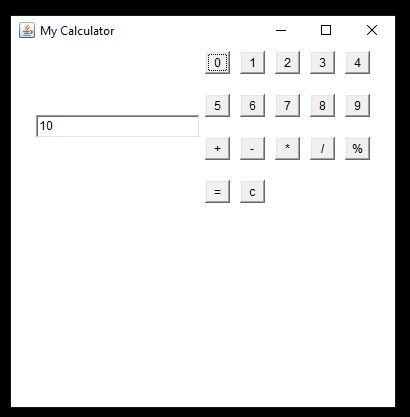
Calculator v=new Calculator ();

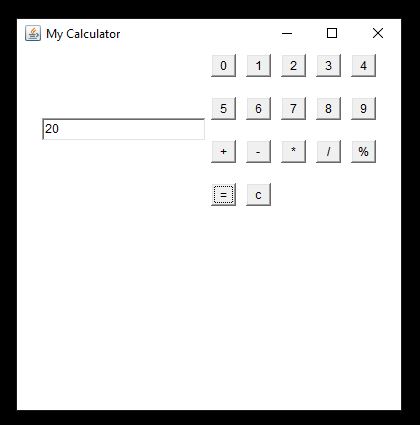
}

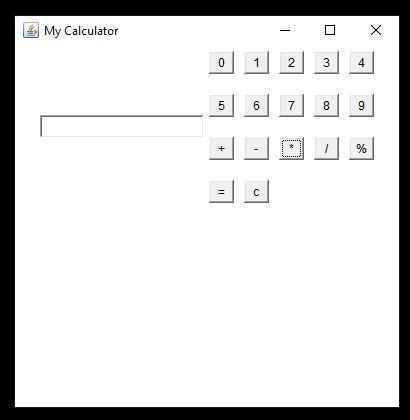
}

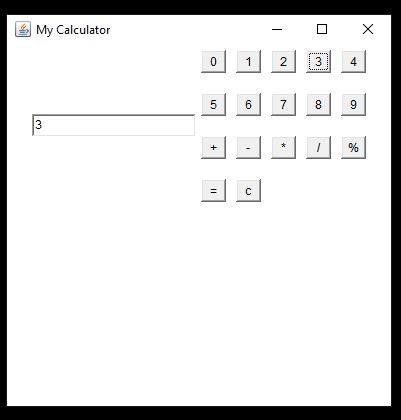
**OUTPUT**

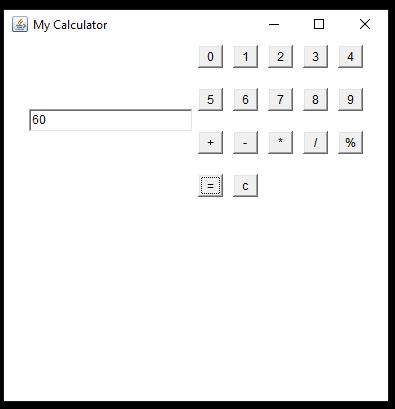
****

****

****

****

****

****

**24.FACE**

import java.io.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.applet.\*;

/\*

<applet code="Face" width=400 height=400>

</applet>

\*/

public class Face extends Applet implements ActionListener

{

String str;

Button happy,sad;

public void init()

{

happy=new Button("happy");

sad=new Button("sad");

add(happy);

add(sad);

happy.addActionListener(this);

sad.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

str=e.getActionCommand();

repaint();

}

public void paint(Graphics g)

{

g.drawOval(40,40,120,150);

g.drawOval(57,75,30,20);

g.drawOval(110,75,30,20);

g.fillOval(68,81,10,10);

g.fillOval(121,81,10,10);

g.drawOval(85,100,30,30);

if(str=="sad")

g.fillArc(70,135,60,30,0,180);

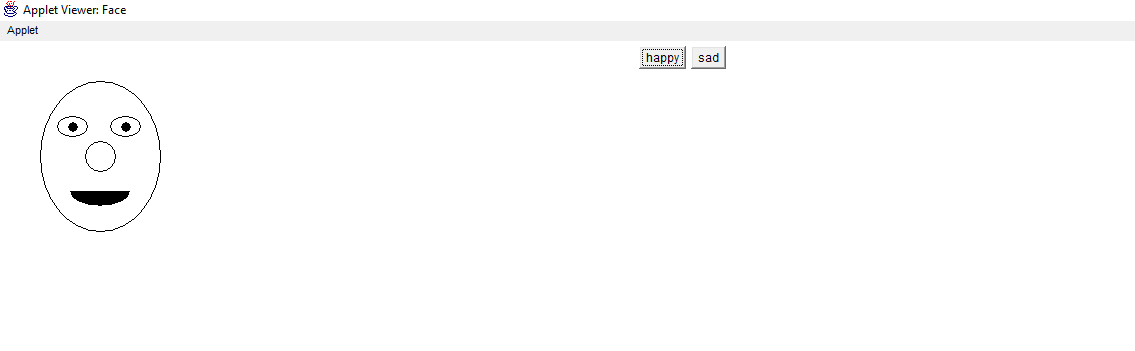
else

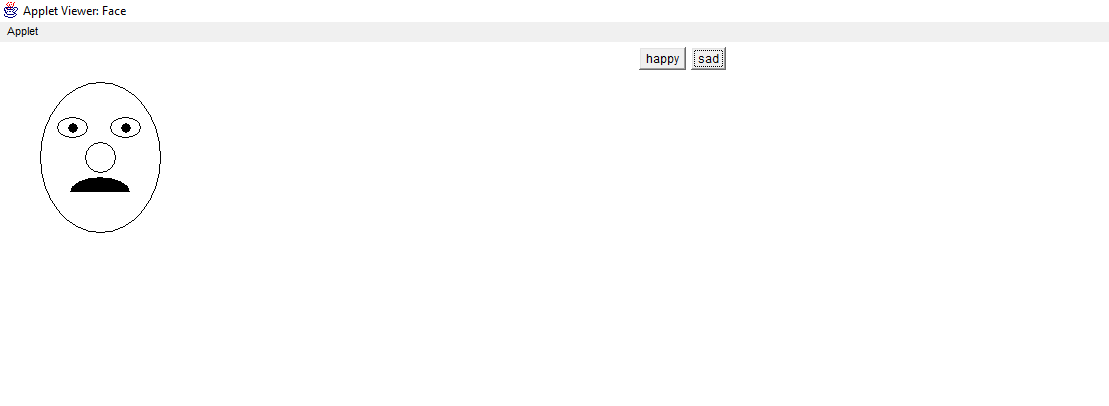
g.fillArc(70,135,60,30,0,-180);

}

}

**OUTPUT**





**25.REMOTE METHOD INVOCATION**

AddServerIntf

import java.rmi.\*;

public interface AddServerIntf extends Remote

{

double add(double d1,double d2)throws RemoteException;

}

AddServerImpl

import java.rmi.\*;

import java.rmi.server.\*;

public class AddServerImpl extends UnicastRemoteObject implements AddServerIntf

{

public AddServerImpl()throws RemoteException

{

}

public double add(double d1,double d2)throws RemoteException

{

return d1+d2;

}

}

AddServer

import java.net.\*;

import java.rmi.\*;

public class AddServer

{

public static void main(String args[])

{

try

{

AddServerImpl addServerImpl=new AddServerImpl();

Naming.rebind("AddServer",addServerImpl);

}

catch(Exception e)

{

System.out.println("Exception"+e);

}}}

AddClient

import java.rmi.\*;

public class AddClient

{

public static void main(String args[])

{

try

{

String addServerURL="rmi://"+args[0]+"/AddServer";

AddServerIntf addServerIntf=(AddServerIntf)Naming.lookup(addServerURL);

System.out.println("the 1st number is"+args[1]);

double d1=Double.valueOf(args[1]).doubleValue();

System.out.println("the 2nd number is"+args[2]);

double d2=Double.valueOf(args[2]).doubleValue();

System.out.println("the sum is:"+addServerIntf.add(d1,d2));

}

catch(Exception e)

{

System.out.println("Exception" +e);

}

}

}

**OUTPUT**

