

JAVA LAB MANUAL

LAB PROGRAMS LIST

1. Write a java program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a,b,c and use the quadratic formula. If the discriminant b^2-4ac is negative, display a message stating that there are not real solutions.
2. The fibonacci sequence is defined by the following rule. The first two values in the sequence are 1 and 1. Every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non recursive functions to print the n^{th} value in the Fibonacci sequence.
3. Write a java program that prompt the user for an integer and then prints out all prime numbers up to that integer.
4. Write a java program that checks whether a given string is a palindrome or not. Ex:MADAM is a palindrome.
5. Write a java program for sorting a given list of names in ascending order.
6. Write a java program to multiply two given matrices.
7. Write a java program that reads a line of integers and then displays each integers, and the sum of all integers (use StringTokenizer class).
8. Write a java program that reads on file name from the user then displays information about whether the file exists, whether the file is readable, whether the file is writable , the type of file and the length of the file bytes.
9. Write a java program that reads a file and displays a file and on the screen , with a line number before each line.
10. Write a java program that displays the number of characters , lines and words in a text file.

11. Write a java program that:
 - a) Implements Stack ADT
 - b) Convert infix expression into postfix form
 12. Write an Applet that displays a simple message.
 13. Write an Applet that computes the payment of a loan based on the amount of the loan , the interest rate and the number of months. It takes one parameter from the browser: Monthly rate; if true, the interest rate is per month; other wise the interest rate is annual.
 14. Write a java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the + - * % operations. Add a text field to display the result.
 15. Write a java program for handling mouse events.
 16. Write a java program for creating multiple threads.
 17. Write a java program that correctly implements producer consumer problem using the concept of inter thread communication.
 18. Write a java program that lets users create Pie charts. Design your own user interface(with swings & AWT).
 19. Write a java program that allows the user to draw lines, rectangles and Ovals.
 20. Write a java program that implements a sample client/server application. The client sends data to a server. The server receives data, uses it to produce a result, and then sends a result back to the client. The client displays the result on the console. For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle.
 21. Write a java program that illustrates how run time polymorphism is achieved.
-
1. Write a java program that prints all real solutions to the quadratic equations $ax^2+bx+c=0$.read in a,b,c and use the quadratic formula .If the discriminate b^2-4ac is negative ,display message stating that there are no real solutions.

CODE

```
import javax.swing.JOptionPane;
import java.text.DecimalFormat;
class lab1
{
    public static void main(String[] args)
```

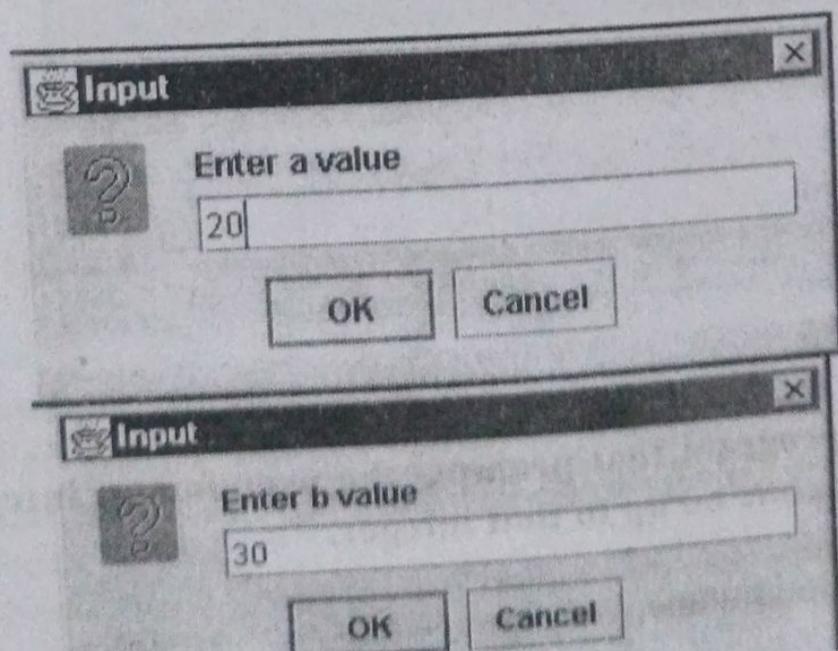
```

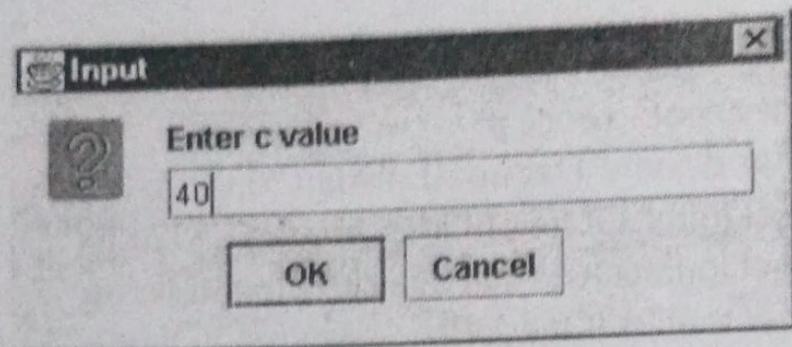
{
double a,b,c,dis,root1,root2,p;
DecimalFormat d=new DecimalFormat("0.00");
a=Double.parseDouble(JOptionPane.showInputDialog("Enter a value"));
b=Double.parseDouble(JOptionPane.showInputDialog("Enter b value"));
c=Double.parseDouble(JOptionPane.showInputDialog("Enter c value"));
dis=(b*b-4*a*c);
if(dis==0)
{
System.out.println("Roots are real and equal");
root1=-b/(2*a);
System.out.println("Root1="+d.format(root1));
System.out.println("Root2="+d.format(root1));
}
else if(dis>0)
{
System.out.println("Roots are real and different");
p=Math.sqrt(dis);
root1=(-b+p)/(2*a);
root2=(-b-p)/(2*a);
System.out.println("Root1="+d.format(root1));
System.out.println("Root2="+d.format(root2));
}
else
System.out.println("Roots are imaginary");
}
}

```

Output::

c:\javac lab1.java
c:\java lab1





C:\ Roots are imaginary

2. The Fibonacci sequence is defined by the following rule the first two values in the sequence are 1 and 1. every subsequent value is the sum of the two values preceding it. Write a java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.

CODE :

```
class lab2
{
    public static void main(String[] args)
    {
        int f0=1,f1=1,i=3,f2,n;
        n=Integer.parseInt(args[0]);
        System.out.println(f0);
        System.out.println(f1);
        while(i<=n)
        {
            f2=f0+f1;
            System.out.println(f2);
            f0=f1;
            f1=f2;
            i++;
        }
    }
}
```

OUTPUT:

```
C:\javac lab2.java
C:\java lab2 10
1 1 2 3 5 8 13 21 34 55
```

3. Write a java program that prompts the user for an integer and then prints out all prime no up to that integer.

CODE

```
import javax.swing.JOptionPane;
```

```

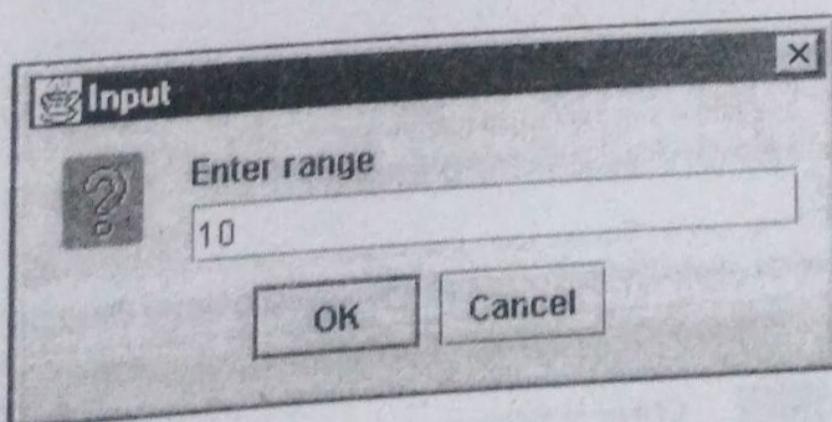
class lab3
{
    public static void main(String[] args)
    {
        int n,j,x,count;
        //n=Integer.parseInt(args[0]);
        n=Integer.parseInt(JOptionPane.showInputDialog("Enter range"));
        for(int i=1;i<=n;i++)
        {
            x=i;
            count=0;
            j=2;
            while(j<x)
            {
                if(x%j==0)
                {
                    count=1;
                    break;
                }
                j++;
            }
            if(count==0)
                System.out.println(x+"is prime");
        }
    }
}

```

OUTPUT:

C:\javac lab3.java

C:\java lab3



C:\1is prime number
 2is prime number
 3is prime number
 5is prime number
 7is prime number

4. A) Write a java program that checks whether a given string is palindrome or not.

Ex. MADAM is a palindrome.

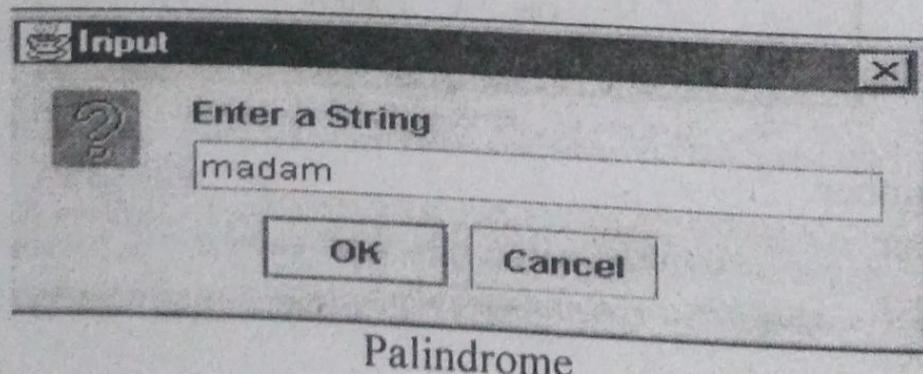
CODE

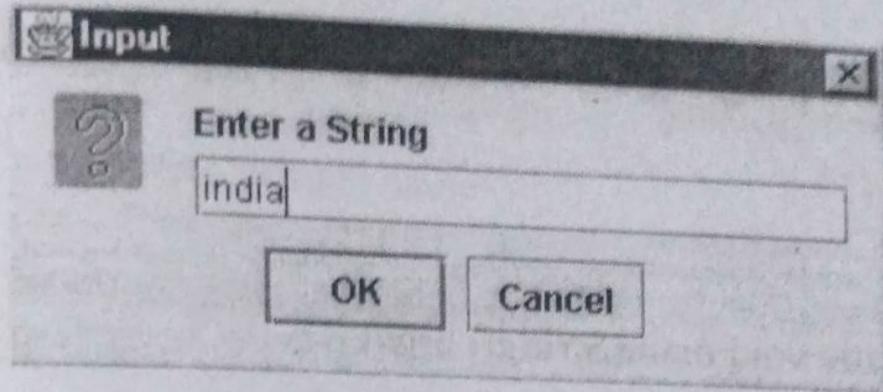
```
import javax.swing.JOptionPane;
class lab4
{
    public static void main(String[] args)
    {
        String s;
        s=JOptionPane.showInputDialog("Enter a String");
        char ch[]=s.toCharArray();
        int i,j,count=0;
        for(i=0,j=ch.length-1;i<j;i++,j--)
        {
            if(ch[i]!=ch[j])
            {
                System.out.println("Not polindrome");
                count=1;
                break;
            }
        }
        if(count==0)
            System.out.println("Polindrome");
    }
}
```

OUTPUT:

C:\javac lab4.java

C:\java lab4





Not polindrome

- B) Example: Write a program to check whether a given number is Armstrong number (153) or not

class labd

```

{
    public static void main(String[] args)
    {
        int x,n,temp,sum=0;
        n=Integer.parseInt(args[0]);
        temp=n;
        do
        {
            x=n%10;
            sum=sum+x*x*x;
            n=n/10;
        }while(n>0);
        if(sum==temp)
            System.out.println("Number is armstrong");
        else
            System.out.println("Number is not armstrong");
    }
}
  
```

OUTPUT:

C:\javac labd.java

C:\java labd 153

Number is Armstrong

C:\java labd 145

Number is not Armstrong

C. Write a java program for sorting a given list of numbers in ascending order.

CODE

```
class sortNum
{
    public static void main(String[] args)
    {
        int temp;
        int no[]={34,78,23,11,99};
        for(int i=0;i<no.length-1;i++)
        {
            for(int j=i+1;j<no.length;j++)
            {
                if(no[i]>no[j])
                {
                    temp=no[i];
                    no[i]=no[j];
                    no[j]=temp;
                }
            }
        }
        for(int i=0;i<no.length;i++)
            System.out.println(no[i]);
    }
}
```

OUTPUT:

```
C:\javac sortNum.java
C:\java sortNum
11 23 34 78 99
```

5. Write a java program for sorting a given list of names in ascending order.

CODE

```
public class sortNames
{
    public static void main(String args[])
    {
        String names[]{"vijay","isaac","chandhu","ramesh"};
        System.out.println("names before sorting");
        Printnames(names);
        Selectionsort(names);
        System.out.println("my list after sort is");
```

```

        Printnames(names);
    }

    static void Printnames(String[] names)
    {
        for(int i=0;i<names.length;i++)
        {
            System.out.println(names[i]\n);
        }
    }

    static void Selectionsort(String[] names)
    {
        String temp;
        int index;
        for(int i=0;i<names.length;i++)
        {
            temp=names[i];
            index=i;
            for(int j=0;j<names.length;j++)
            {
                int k=names[i].compareTo(names[j]);
                if(k>0)
                {
                    temp=names[i];
                    names[i]=names[j];
                    names[j]=temp;
                }
            }
        }
    }
}

```

OUTPUT:

C:\>javac sortNames.java

C:\>java sortName

Chandhu

Isaac

Ramesh

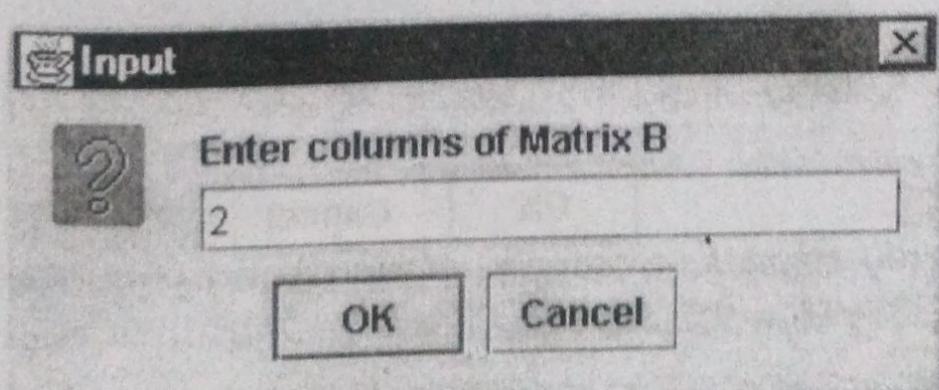
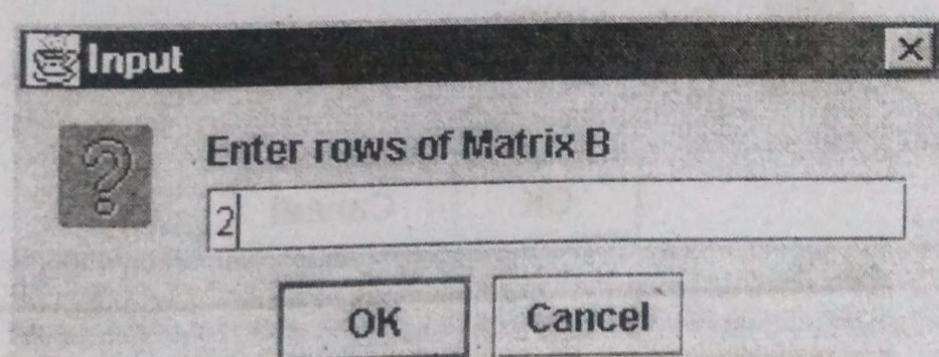
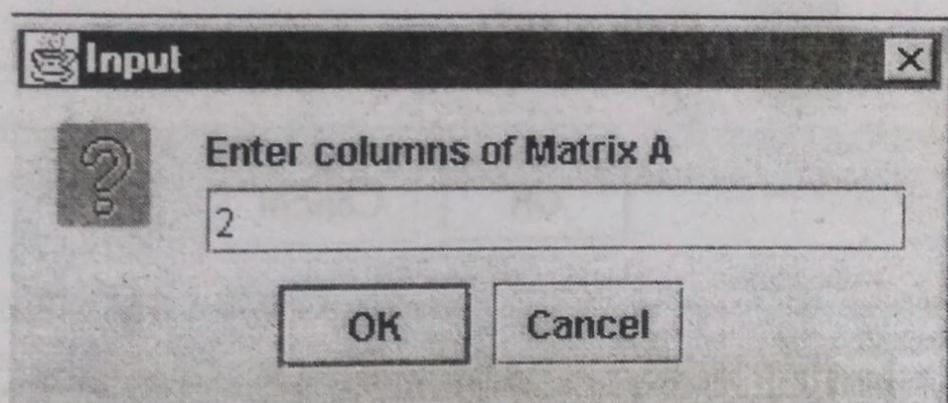
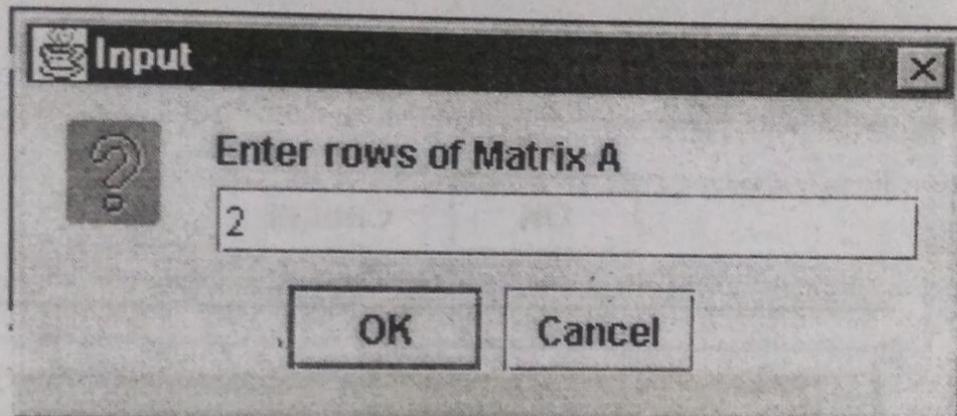
Vijay

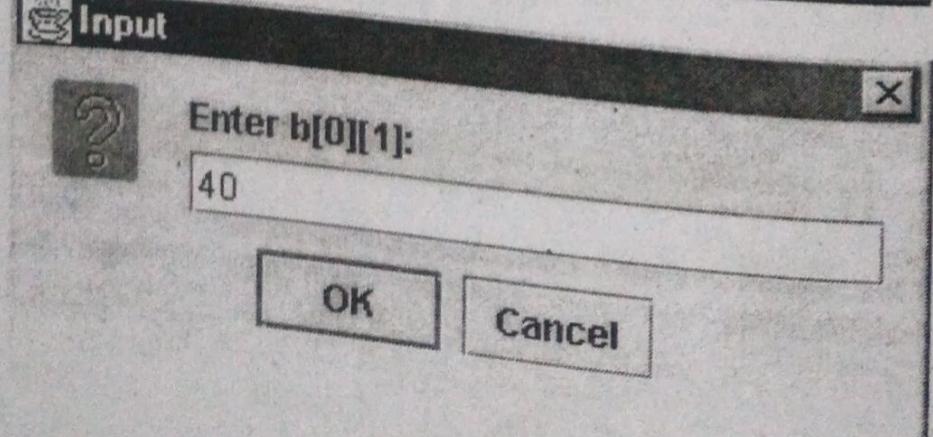
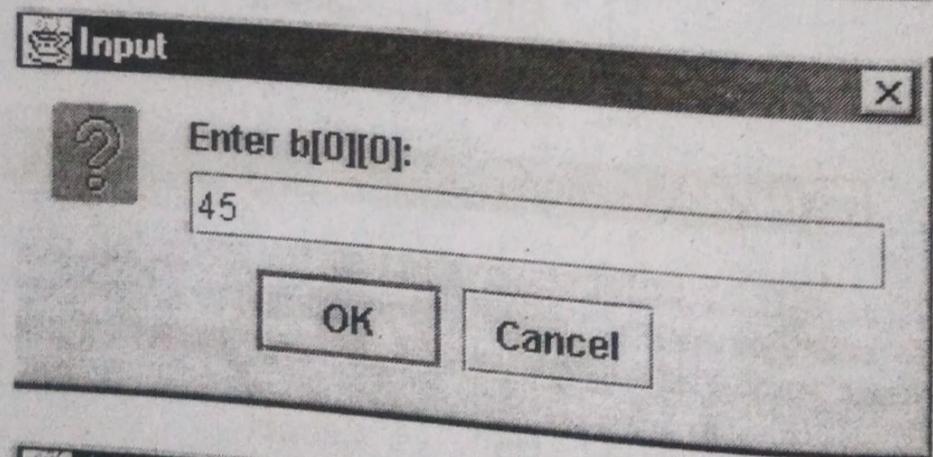
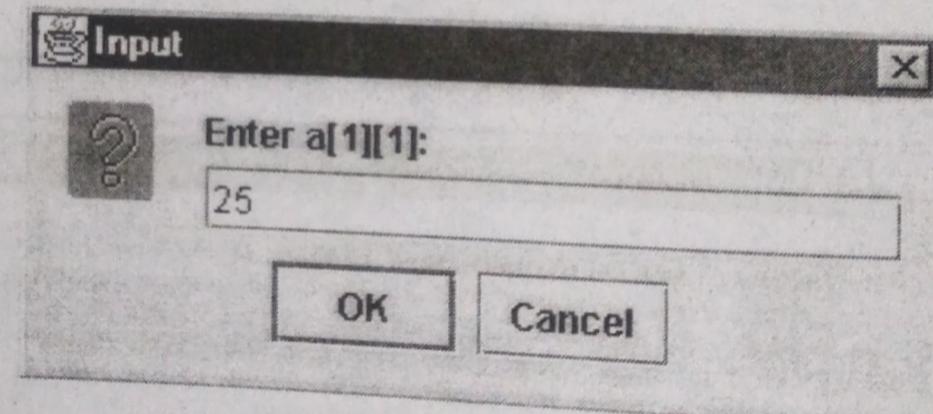
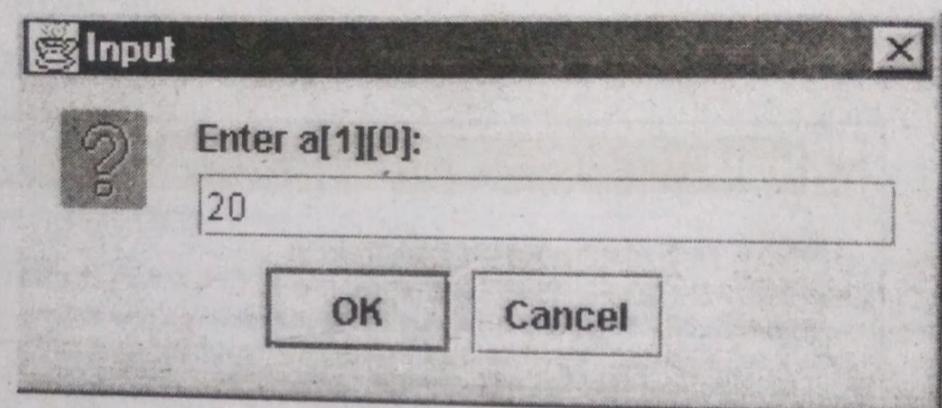
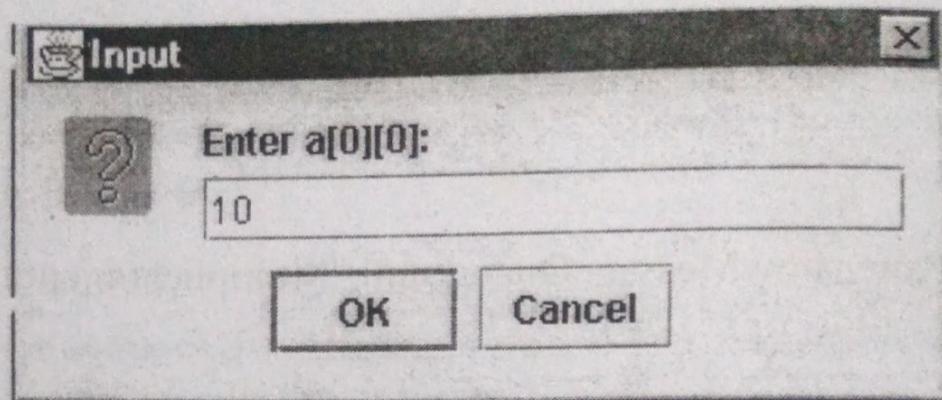
6. Write a java program to multiply two given matrices.**CODE**

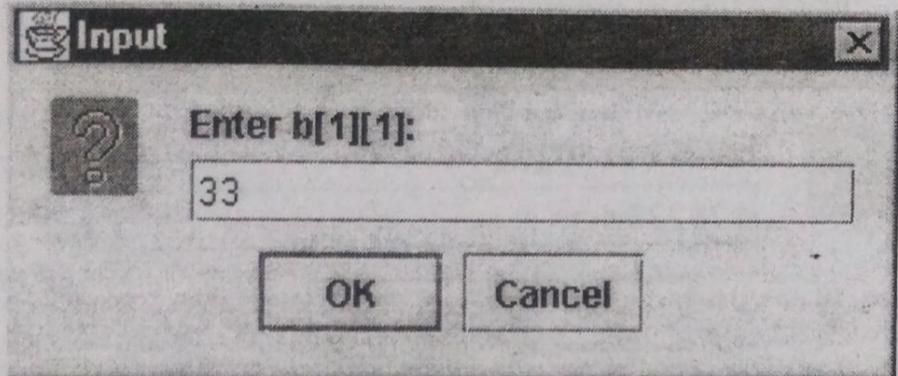
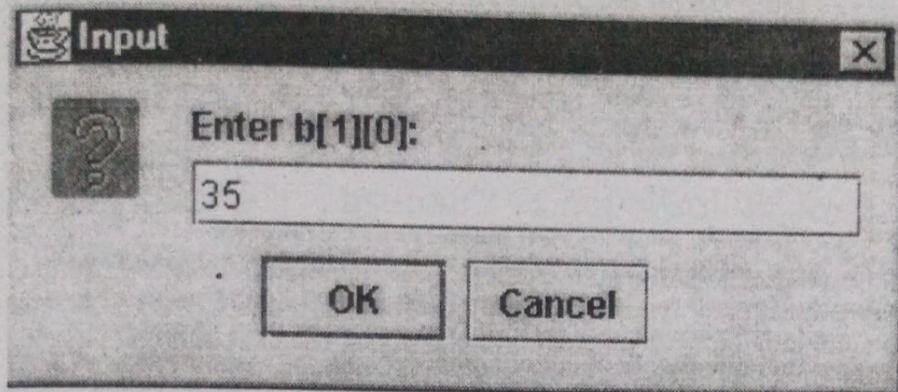
```
import javax.swing.JOptionPane;
class arr2
{
    public static void main(String[] args)
    {
        int m,n,p,q,i,j,k;
        int a[][]=new int [10][10];
        int b[][]=new int [10][10];
        int c[][]=new int [10][10];
        m=Integer.parseInt(JOptionPane.showInputDialog("Enter rows of Matrix A"));
        n=Integer.parseInt(JOptionPane.showInputDialog("Enter columns of Matrix A"));
        p=Integer.parseInt(JOptionPane.showInputDialog("Enter rows of Matrix B"));
        q=Integer.parseInt(JOptionPane.showInputDialog("Enter columns of Matrix B"));
        if(n==p)
        {
            for(i=0;i<m;i++)
            {
                for(j=0;j<n;j++)
                    a[i][j]=Integer.parseInt(JOptionPane.showInputDialog("Enter
a["+i+"]["+j+"]:"));
            }
            for(i=0;i<p;i++)
            {
                for(j=0;j<q;j++)
                    b[i][j]=Integer.parseInt(JOptionPane.showInputDialog("Enter
b["+i+"]["+j+"]:"));
            }
            for(i=0;i<m;i++)
            {
                for(j=0;j<q;j++)
                {
                    c[i][j]=0;
                    for(k=0;k<n;k++)
                        c[i][j]=c[i][j]+a[i][k]*b[k][j];
                }
            }
            for(i=0;i<m;i++)
            {
                System.out.println();
                for(j=0;j<q;j++)
                    System.out.print(" "+c[i][j]);
            }
        }
    }
}
```

```
        else
    {
        JOptionPane.showMessageDialog(null,"Multiplication not possible");
    }
}
```

OUTPUT:







975 895
1775 1625

7. Write a java program that reads a line of integers, and then displays each integer, and the sum of all integers (using StringTokenizer class).

CODE

```
import javax.swing.JOptionPane;
import java.util.*;
class lab7
{
    public static void main(String[] args)
    {
        //String s=new String("4$8$9$67");
        //StringTokenizer t=new StringTokenizer(s,"$");
        String s;
        s=JOptionPane.showInputDialog("Enter the string");
        StringTokenizer t=new StringTokenizer(s,"$");
        int a,sum=0;
        System.out.println(t.countTokens());
        while(t.hasMoreTokens())
        {
            //a=Integer.parseInt(t.nextToken());
            System.out.println(t.nextToken());
        }
    }
}
```

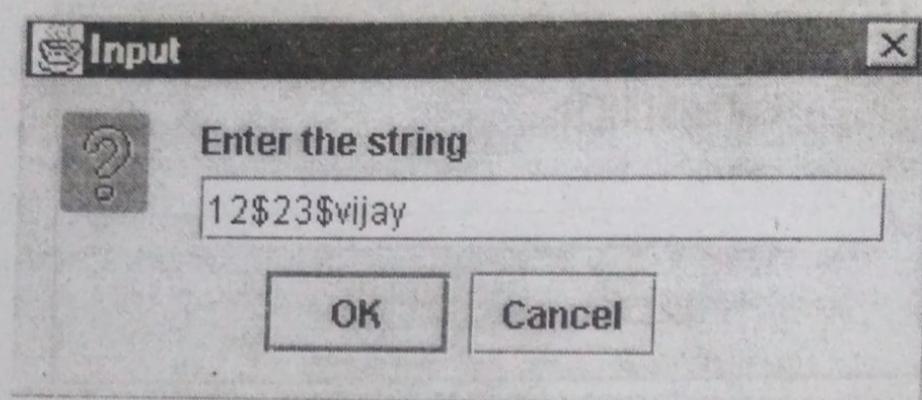
```

        sum=sum+1;
    }
    System.out.println("Total="+sum);
}
}

```

OUTPUT:

C:\502>java lab7



3
12
23
vijay
Total=3

8. Write a java program that reads on file name from the user then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.

CODE

```

import java.io.File;
class FileDemo
{
    static void p(String s)
    {
        System.out.println(s);
    }
    public static void main(String args[])
    {
        File f1=new File("durga.txt");
        p("FileName:"+f1.getName());
        p("Path:"+f1.getPath());
        p("AbsPath:"+f1.getAbsoluteFilePath());
        p("Parent:"+f1.getParent());
        p(f1.exists()?"exists":"does not exist");
    }
}

```

```

p(f1.canWrite()?"is writable":"is not writeable");
p(f1.canRead()?"is readable":"is not readable");
p("is"+(f1.isDirectory()?"":"not"+"a directory"));
p(f1.isFile()?"is normal file":"might be a named pipe");
p(f1.isAbsolute()?"is absolute":"is not absolute");
p("File last modified:"+f1.lastModified());
p("File size:"+f1.length()+"Bytes");
}
}

```

OUTPUT

C:\502>javac FileDemo.java

C:\502>java FileDemo

FileName:durga.txt

Path:durga.txt

AbsPath:C:\502\durga.txt

Parent:null

does not exist

is not writeable

is not readable

is not a directory

might be a named pipe

is not absolute

File last modified:0

File size:0Bytes

9. Write a java program that reads a file displays a file on the screen, with a line no before each line.

```

import java.io.*;
class charArrayWriterdemo
{
    public static void main(String[] args) throws IOException
    {
        CharArrayWriter f=new CharArrayWriter();
        String s="this should end up in the array";
        char buf[]=new char[s.length()];
        s.getChars(0,s.length(),buf,0);
    }
}

```

```

        f.write(buf);
        System.out.println("buffer as string");
        System.out.println("f.toString()");
        System.out.println("intoarray");
        char c[]=f.toCharArray();
        for(int i=0;i<c.length;i++)
        {
            System.out.println(c[i]);
        }
        System.out.println("\n to a file writer()");
        FileWriter f2=new FileWriter("test.txt");
        f.writeTo(f2);
        f2.close();
        System.out.println("doing a reset");
        f.reset();
        for(int i=0;i<3;i++)
        f.write('x');
        System.out.println(f.toString());
    }
}

```

OUTPUT:

C:\502>javac charArrayWriterdemo.java

C:\502>java charArrayWriterdemo

buffer as string

f.toString()

intoarray

this should end up in the array

to a file writer()

doing a reset

xxx

- 10. Write a java program that displays no of characters, lines and words in a text file.**

import java.io.*;

```
class wordcount
{
    public static int words=0;
    public static int lines=0;
    public static int chars=0;
    public static void wc(InputStreamReader isr) throws IOException
    {
        int c=0;
        boolean lastwhite=true;
        String whitespace="\t\n\r";
        while((c=isr.read())!=-1)
        {
            chars++;
            if(c=='\n')
            {
                lines++;
            }
            int index=whitespace.indexOf(c);
            if(index==-1)
            {
                if(lastwhite==true)
                {
                    ++words;
                }
                lastwhite=false;
            }
            else
            {
                lastwhite=true;
            }
            if(chars!=0)
            {
                ++lines;
            }
        }
    }
}
```

```
}

public static void main(String args[])
{
    FileReader fr;
    try
    {
        if(args.length==0)
        {
            wc(new InputStreamReader(System.in));
        }
        else
        {
            for(int i=0;i<args.length;i++)
            {
                fr=new FileReader(args[i]);
                wc(fr);
            }
        }
    }
    catch(IOException e)
    {
        return;
    }
    System.out.println(lines+" "+words+" "+chars);
}
}
```

INPUT:

This is OOPS through JAVA

It is a pure object oriented programming

OUTPUT:

No of lines: 2

No of words :12

No of Characters:65

11. Write a java program that Implements stack ADT

```

import java.io.*;
import java.lang.*;
class StackDemo {
    public static void main(String args[]) {
        Stack mystack1 = new Stack();
        Stack mystack2 = new Stack();

        // push some numbers onto the stack
        for(int i=0; i<10; i++) mystack1.push(i);
        for(int i=10; i<20; i++) mystack2.push(i);

        // pop those numbers off the stack
        System.out.println("Stack in mystack1:");
        for(int i=0; i<10; i++)
            System.out.println(mystack1.pop());

        System.out.println("Stack in mystack2:");
        for(int i=0; i<10; i++)
            System.out.println(mystack2.pop());

        // these statements are not legal
        // mystack1.tos = -2;
        // mystack2.stck[3] = 100;
    }
}

```

12. Write an applet that displays a simple message.

```

<HTML>
<applet code="message.class" height=100 width=100>
</applet>
</HTML>/*

```

```

import java.awt.*;
import java.applet.*;
public class message extends Applet
{
    public void paint(Graphics g)
    {
        g.drawString("Merits",50,50);
    }
}

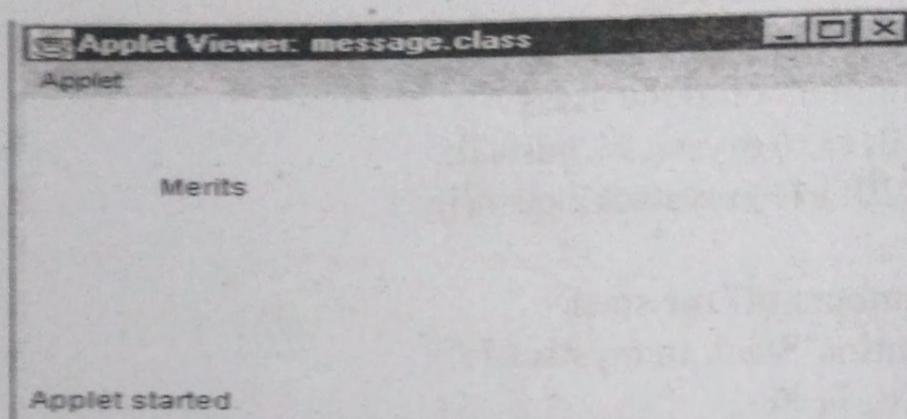
```

OUTPUT

C:\502>javac message.java

C:\502>appletviewer message

C:\502>appletviewer message.java



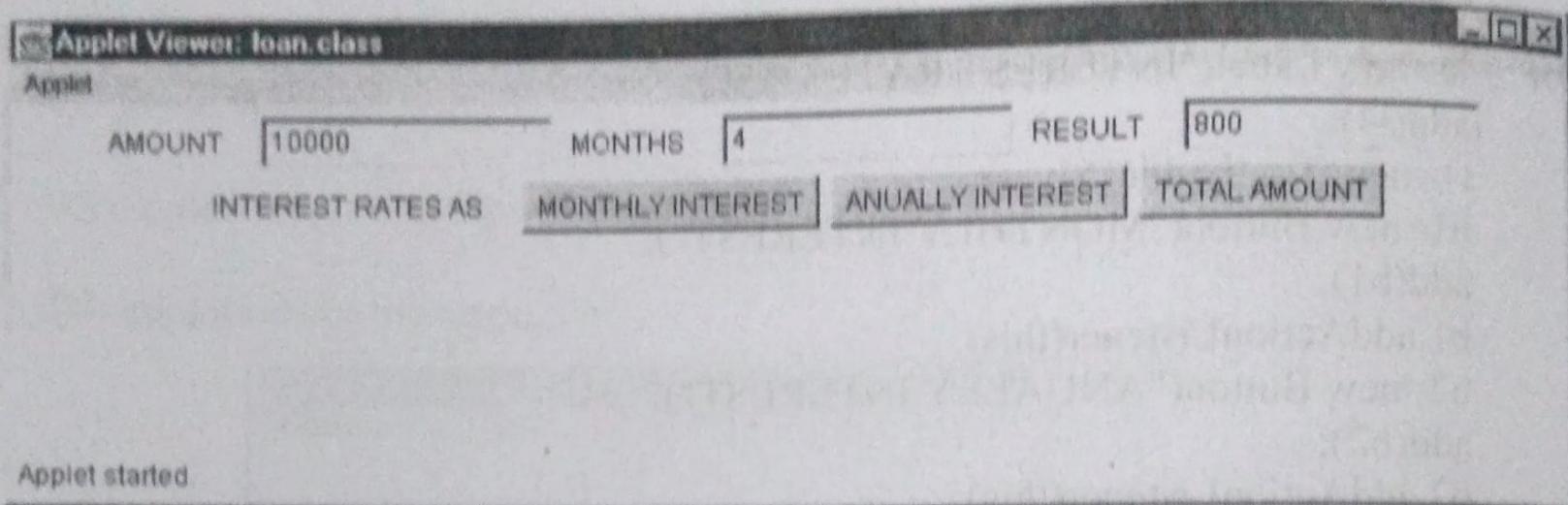
13. Write an applet that computes the payment of a loan based on the amount of the loan, the interest rate and the no of months it takes one parameter from browser: monthly rate; if true ,the interest rate is per month other wise the interest rate is annual.

CODE

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="loan.class" width=300 height=300>
</applet>*
public class loan extends Applet implements ActionListener
{
Label l1,l2,l3,l4;
TextField t1,t2,t3;
Button b1,b2,b3,b4;
public void init()
{
l1=new Label("AMOUNT");
add(l1);
t1=new TextField(15);
add(t1);
l2=new Label("MONTHS");
add(l2);
t2=new TextField(15);
add(t2);
l3=new Label("RESULT");
add(l3);
t3=new TextField(15);
```

```
add(t3);
l4=new Label("INTEREST RATES AS");
add(l4);
t1=new TextField(15);
b1=new Button("MONTHLY INTEREST");
add(b1);
b1.addActionListener(this);
b2=new Button("ANUALLY INTEREST");
add(b2);
b2.addActionListener(this);
b3=new Button("TOTAL AMOUNT");
add(b3);
b3.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
if(ae.getSource()==b1)
{
int a=Integer.parseInt(t1.getText());
int m=Integer.parseInt(t2.getText());
double rate=(a*m*2)/100.0;
t3.setText(""+rate);
}
if(ae.getSource()==b2)
{
int a=Integer.parseInt(t1.getText());
int m=Integer.parseInt(t2.getText());
double rate=(a*m*15)/(12*100.0);

t3.setText(""+rate);
}
if(ae.getSource()==b1)
{
int a=Integer.parseInt(t1.getText());
int m=Integer.parseInt(t2.getText());
double rate=(a*m*2)/100.0;
rate=rate+a;
t3.setText(""+rate);
}
}
}
}
//gvsreenu@yahoo.com, rajendra.c@rediffmail.com
```



14. Write a java program that works as a simple calculator, use a grid layout to arrange buttons for the digits and for the + - * % operations. Add a text field to display the result.

```
import java.io.*;
import java.applet.Applet;
import java.awt.event.*;
/* <applet code ="Buttontdemo.class" height=300 width=300>
</applet>*
public class Buttontdemo extends Applet implements ActionListener
{
    TextField t1,t2,t3;
    Button b1,b2,b3;
    Label l1,l2,l3;
    String msg="";
    Public void init()
    {
        l1=new Label("enter the first number");
        add(l1);
        t1=new TextField(15);
        add(t1);
        l2=new Label("enter the second number");
        add(l2);
        t2=new TextField(15);
        add(t2);
        l3=new Label("enter the third number");
        add(l3);
        t3=new TextField(15);
        add(t3);
        b1=new Button("ADD");
        add(b1);
        b1.addActionListener(this);
    }
}
```

```

b2=new Button("SUB");
add(b2);
b2.addActionListener(this);

b3=new Button("MUL");
add(b3);
b3.addActionListener(this);

b4=new Button("DIV");
add(b4);
b4.addActionListener(this);
}

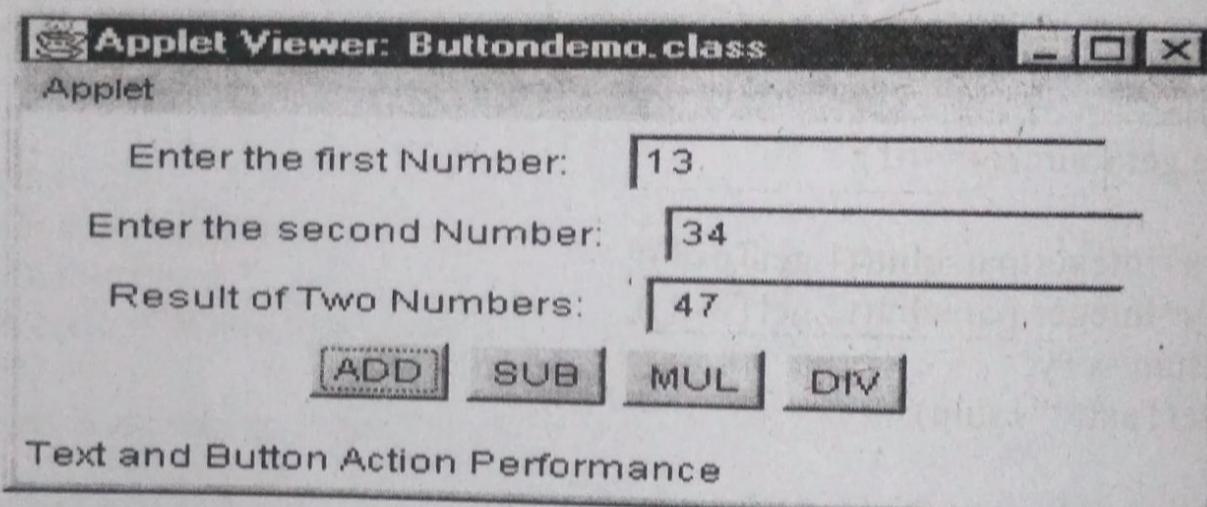
Public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b1)
{
int x=Integer.parseInt(t1.getText());
int y=Integer.parseInt(t2.getText());
int sum=x+y;
t3.setText(""+sum);
}
Public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b4)
{
int x=Integer.parseInt(t1.getText());
int y=Integer.parseInt(t2.getText());
int div=x/y;
t3.setText(""+div);
}
Public void actionPerformed(ActionEvent e)
{
if(e.getSource()==b2)
{
int x=Integer.parseInt(t1.getText());
int y=Integer.parseInt(t2.getText());
int sub=x-y;
t3.setText(""+sub);
}
showStatus("Text and Button Action Performance");
repaint();
}
Public void actionPerformed(ActionEvent e)

```

```

{
if(e.getSource()==b3)
{
int x=Integer.parseInt(t1.getText());
int y=Integer.parseInt(t2.getText());
int mul=x*y;
t3.setText(""+mul);
}
showStatus("Text and Button Action Performance");
repaint();
}

```

OUTPUT:**15. Write a java program for handling mouse events.****Program code**

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
/*<applet code="mouseevents.class" width=300 height=300>
</applet>*/
public class mouseevents extends Applet implements
MouseListener,MouseMotionListener
{
    String msg="";
    int mouseX=0,mouseY=0;
    public void init()
    {
        addMouseListener(this);
        addMouseMotionListener(this);
    }
    public void mouseClicked(MouseEvent me)
    {
        mouseX=0;
    }
}

```

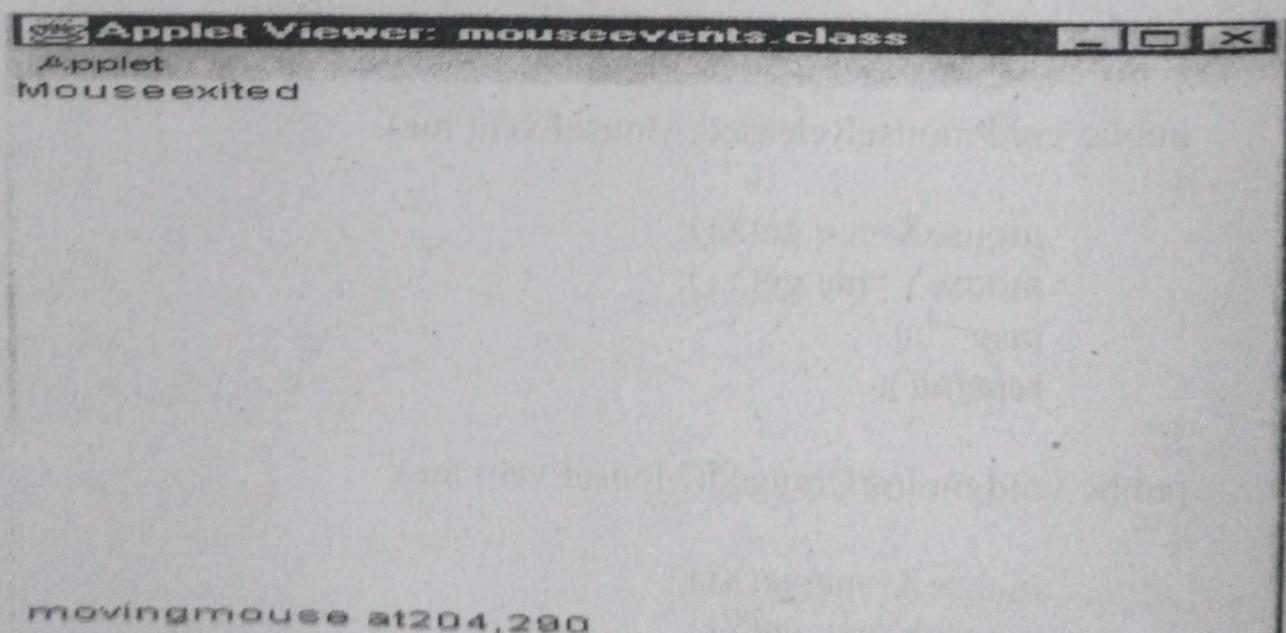
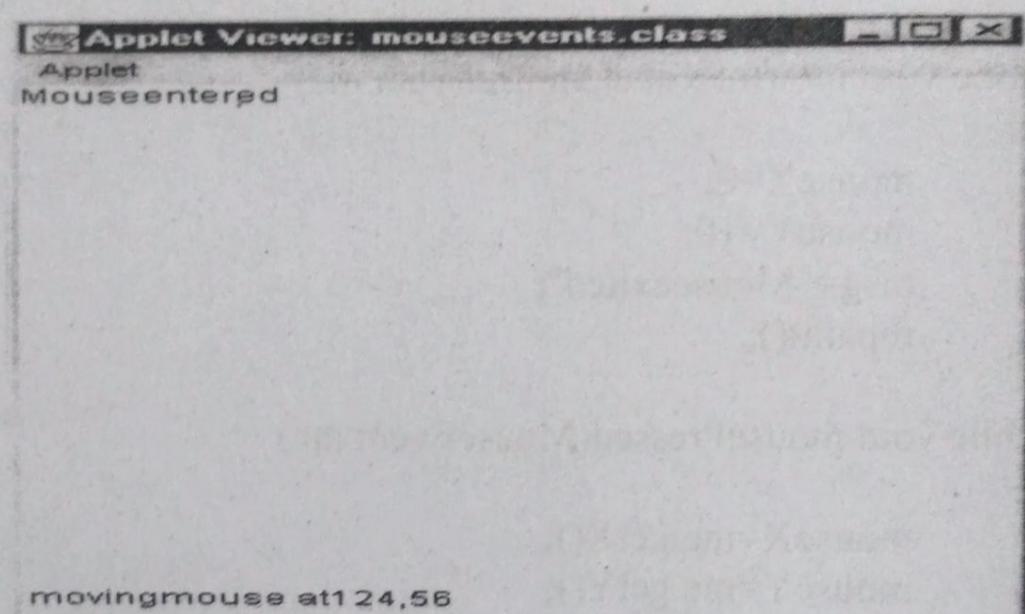
```
mouseY=10;
msg="Mouseclicked";
repaint();
}
public void mouseEntered(MouseEvent me)
{
    mouseX=0;
    mouseY=10;
    msg="Mouseentered";
    repaint();
}
public void mouseExited(MouseEvent me)
{
    mouseX=0;
    mouseY=10;
    msg="Mouseexited";
    repaint();
}
public void mousePressed(MouseEvent me)
{
    mouseX=me.getX();
    mouseY=me.getY();
    msg="down";
    repaint();
}
public void mouseReleased(MouseEvent me)
{
    mouseX=me.getX();
    mouseY=me.getY();
    msg="up";
    repaint();
}
public void mouseDragged(MouseEvent me)
{
    mouseX=me.getX();
    mouseY=me.getY();
    msg="*";
    showStatus("dragging mouse at"+mouseX+","+mouseY);
    repaint();
}
public void mouseMoved(MouseEvent me)
{
    showStatus("moving mouse at"+me.getX()+","+me.getY());
```

```
    }
    public void paint(Graphics g)
    {
        g.drawString(msg,mouseX,mouseY);
    }
}
```

OUTPUT:

C:\502>javac mouseevents.java

C:\502>appletviewer mouseevents.java



16. Write a java program for creating multiple threads.
Program code

class thread1 extends Thread

{

public void run()

{

```

try
{
    for(int i=1;i<=10;i++)
    {
        System.out.println("\t\n"+i);
        Thread.sleep(1000);
    }
}
catch(Exception e)
{
    System.out.println("exception");
}

}

class thread2 extends Thread
{
    public void run()
    {
        try
        {
            for(int i=1;i<=10;i++)
            {
                System.out.println("RAM");
                Thread.sleep(1000);
            }
        }
        catch(Exception e)
        {
            System.out.println("exception");
        }
    }
}

class RAM10
{
    public static void main(String args[])
    {
        thread1 t1=new thread1();
        thread2 t2=new thread2();
        t1.start();
        t2.start();
    }
}

```

OUTPUT:

C:\502>javac RAM10.java

C:\502>java RAM10

1
RAM
2
RAM
3
RAM
4
RAM
5
RAM
6
RAM
7
RAM
8
RAM
9
RAM
10
RAM

17. Write a java program that correctly implements producer consumer problem using the concept of inter thread communication.

Program code::

```
class Q
{
    int n;
    boolean valueset=false;
    synchronized int get()
    {
        if(!valueset)
            try
            {
                wait();
            }
            catch(InterruptedException e)
            {
                System.out.println("Exception");
            }
    }
}
```

```

        }

        System.out.println("get"+n);
        valueset=false;
        notify();
        return n;
    }

    synchronized int put(int n)
    {
        if(valueset)
            try
            {
                wait();
            }
            catch(InterruptedException e)
            {
                System.out.println("exception2");
            }
        this.n=n;
        valueset=true;
        System.out.println("put"+n);
        notify();
        return n;
    }
}

class producer implements Runnable
{
    Q p;
    producer(Q p)
    {
        this.p=p;
        new Thread(this,"producer").start();
    }
    public void run()
    {
        int i=0;
        while(true)
        {
            p.put(i++);
        }
    }
}

```

```

class consumer implements Runnable

```

```
{  
    Q q;  
    consumer(Q q)  
    {  
        this.q=q;  
        new Thread(this,"consumer").start();  
    }  
    public void run()  
    {  
        while(true)  
        {  
            q.get();  
        }  
    }  
}  
class pc  
{  
    public static void main(String args[])  
    {  
        Q q=new Q();  
        new producer(q);  
        new consumer(q);  
        System.out.println("press control c to exit");  
    }  
}
```

OUTPUT:

c:\>502>javac pc.java

c:\>502>java pc

get31807

put31808

get31808

put31809

get31809

put31810

get31810

put31811

get31811

put31812

get31812

put31813

get31813

put31814

get31814
put31815
get31815
put31816
get31816
put31817
get31817
put31818

18. Write a java program that lets user create Pie charts. Design your own user interface (with swings and AWT).

```
import java.awt.*;  
import java.awt.event.*;  
import java.applet.*;  
  
public class ArcTest extends Applet {  
    ArcControls controls; // The controls for marking and filling arcs  
    ArcCanvas canvas; // The drawing area to display arcs  
  
    public void init() {  
        setLayout(new BorderLayout());  
        canvas = new ArcCanvas();  
        add("Center", canvas);  
        add("South", controls = new ArcControls(canvas));  
    }  
  
    public void destroy() {  
        remove(controls);  
        remove(canvas);  
    }  
  
    public void start() {  
        controls.setEnabled(true);  
    }  
  
    public void stop() {  
        controls.setEnabled(false);  
    }  
  
    public void processEvent(AWTEvent e) {  
        if (e.getID() == Event.WINDOW_DESTROY) {
```

```

        System.exit(0);
    }
}

public static void main(String args[]) {
    Frame f = new Frame("ArcTest");
    ArcTestarcTest = new ArcTest();

    arcTest.init();
    arcTest.start();

    f.add("Center", arcTest);
    f.setSize(300, 300);
    f.show();
}

public String getAppletInfo() {
    return "An interactive test of the Graphics.drawArc and \nGraphics.fillArc routines.
    Can be run \neither as a standalone application by typing 'java
ArcTest'
    \nor as an applet in the AppletViewer.";
}

class ArcCanvas extends Canvas {
    int startAngle = 0;
    int endAngle = 45;
    boolean filled = false;
    Font font;

    public void paint(Graphics g) {
        Rectangle r = getBounds();
        int hlines = r.height / 10;
        int vlines = r.width / 10;

        g.setColor(Color.pink);
        for (int i = 1; i <= hlines; i++) {
            g.drawLine(0, i * 10, r.width, i * 10);
        }
        for (int i = 1; i <= vlines; i++) {
            g.drawLine(i * 10, 0, i * 10, r.height);
        }
    }
}

```

```

g.setColor(Color.red);
if(filled) {
    g.fillArc(0, 0, r.width - 1, r.height - 1, startAngle, endAngle);
} else {
    g.drawArc(0, 0, r.width - 1, r.height - 1, startAngle, endAngle);
}

```

```

g.setColor(Color.black);
g.setFont(font);
g.drawLine(0, r.height / 2, r.width, r.height / 2);
g.drawLine(r.width / 2, 0, r.width / 2, r.height);
g.drawLine(0, 0, r.width, r.height);
g.drawLine(r.width, 0, 0, r.height);
int sx = 10;
int sy = r.height - 28;
g.drawString("S = " + startAngle, sx, sy);
g.drawString("E = " + endAngle, sx, sy + 14);
}

```

```

public void redraw(boolean filled, int start, int end) {
    this.filled = filled;
    this.startAngle = start;
    this.endAngle = end;
    repaint();
}
}

```

```

class ArcControls extends Panel
    implements ActionListener {
    TextField s;
    TextField e;
    ArcCanvas canvas;

public ArcControls(ArcCanvas canvas) {
    Button b = null;
    this.canvas = canvas;
}

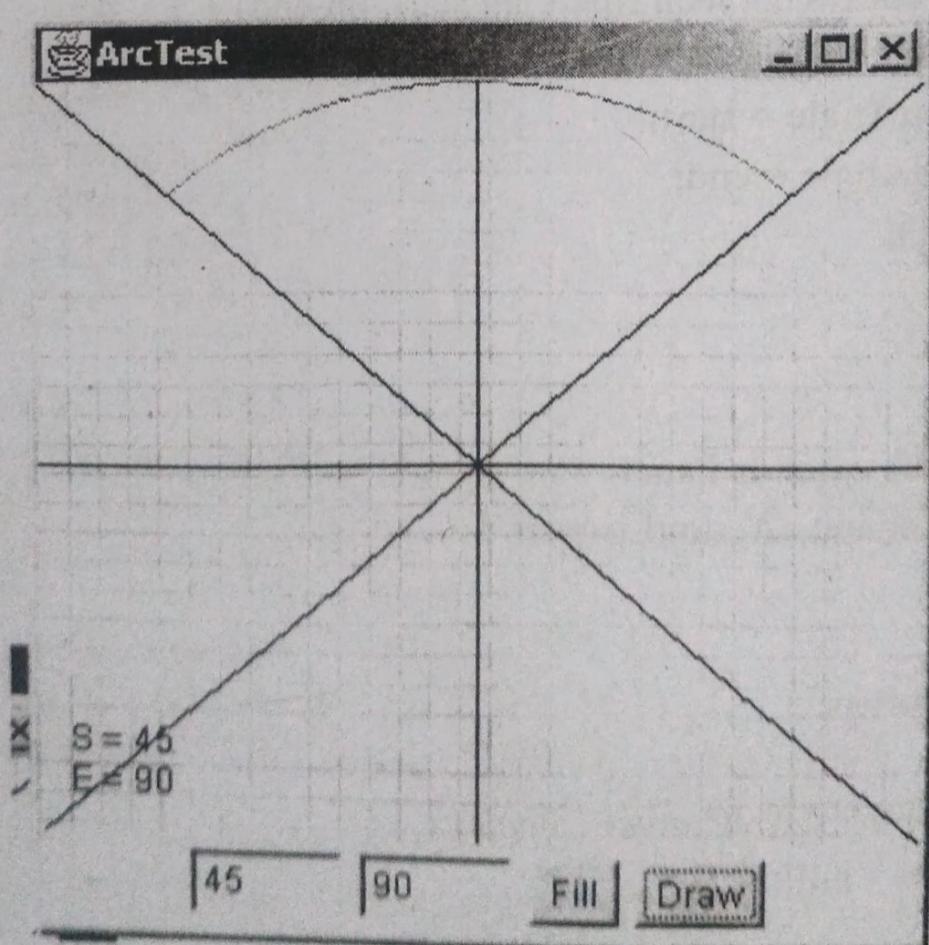
```

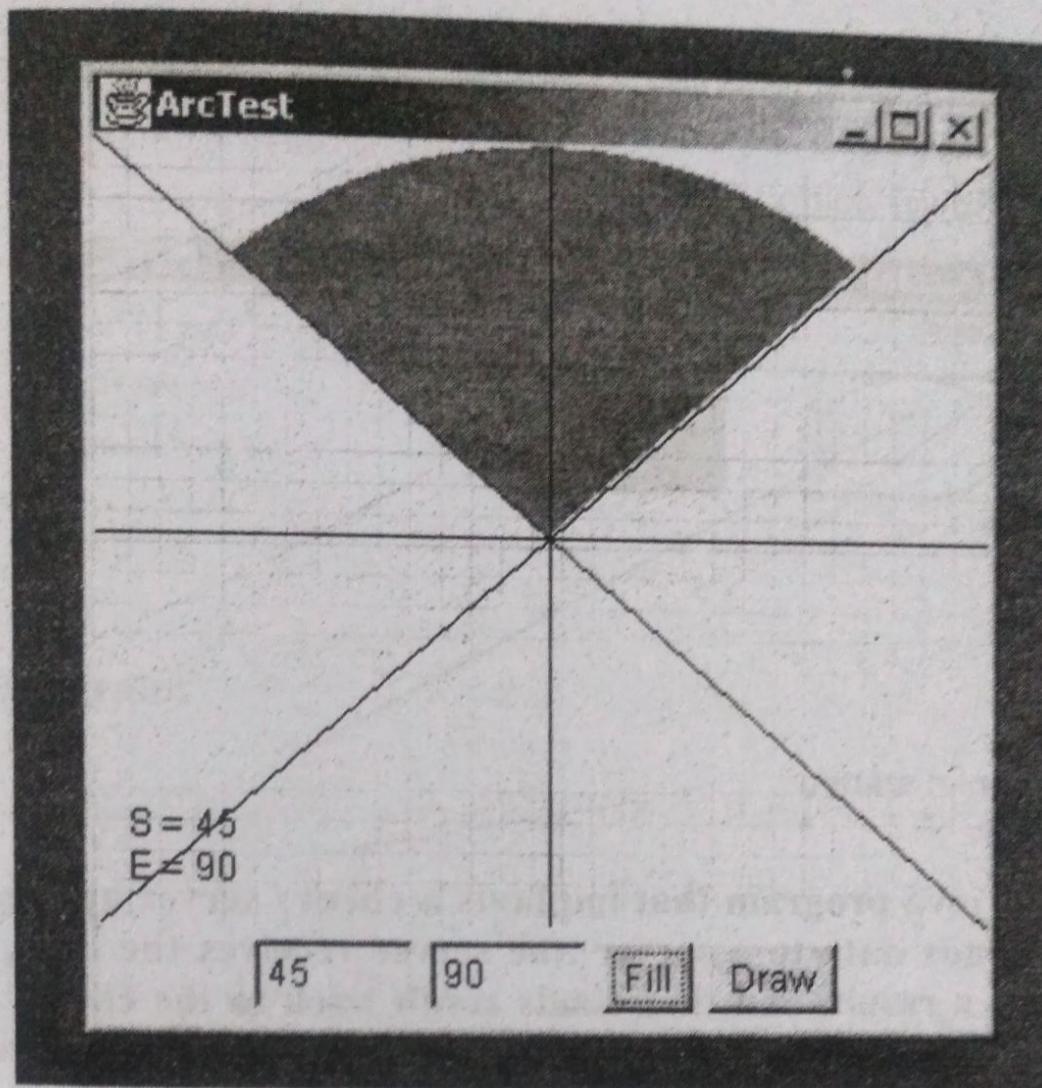
```
add(s = new TextField("0", 4));
add(e = new TextField("45", 4));
b = new Button("Fill");
b.addActionListener(this);
add(b);
b = new Button("Draw");
b.addActionListener(this);
add(b);
}

public void actionPerformed(ActionEvent ev) {
    String label = ev.getActionCommand();

    canvas.redraw(label.equals("Fill"),
                  Integer.parseInt(s.getText().trim()),
                  Integer.parseInt(e.getText().trim()));
}
}
```

o/p





19. Write a java program that allows the user to draw lines, rectangles and Ovals.

Program code

```
/*<html>
<applet code="demo.class" width=300 height=300>
</applet>
</html>*/
```

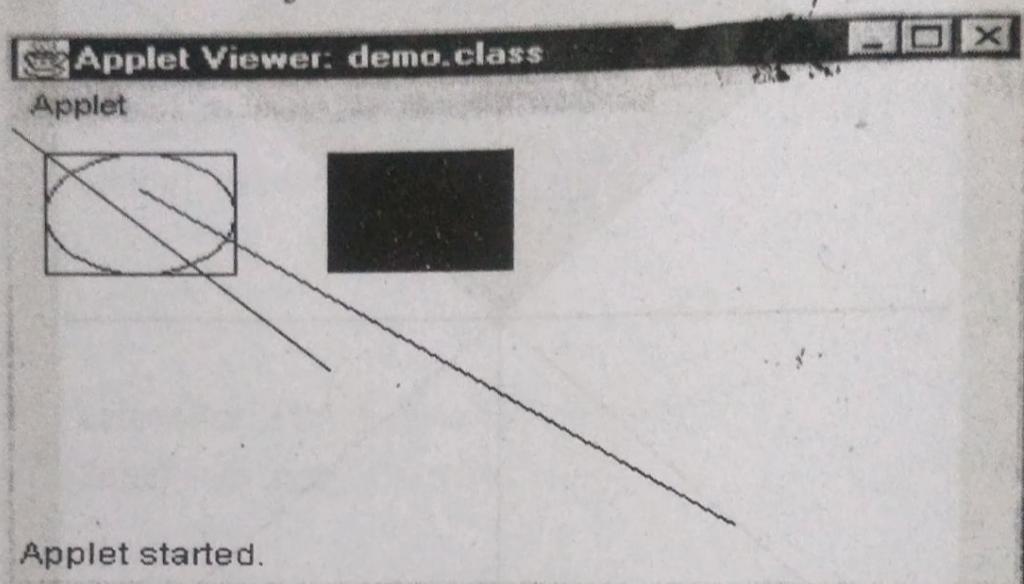
```
import java.awt.*;
import java.applet.*;
public class demo extends Applet
```

```
{
    public void paint(Graphics g)
    {
        g.drawLine(0,0,100,100);
        g.drawLine(40,25,250,180);
        g.drawRect(10,10,60,50);
        g.fillRect(100,10,60,50);
        g.drawOval(10,10,60,50);
        g.fillOval(100,10,60,50);
    }
}
```

OUTPUT:

C:\502>javac demo.java

C:\502>appletviewer demo.java



20. Write a java program that implants a client / server application. The client sends data to a server .the server receives the data, uses it to produce a result, and then sends result back to the client. The client displays the result on t he console .for Ex. the data sent from the client is the radius of the circle, and the result produce by the server is the area of the circle.

Server side program:

```
import java.io.*;
import java.net.*;
class server
{
    public static void main(String args[]) throws Exception
    {
        ServerSocket ss=new ServerSocket(8080);
        System.out.println("wait for client request");
        Socket s=ss.accept();
        BufferedReader br;
        PrintStream ps;
        String str;
        br=new BufferedReader(new
InputStreamReader(s.getInputStream()));
        str=br.readLine();
        System.out.println("received radius");
```

```

        double r=Double.parseDouble(str);
        double area=3.14*r*r;
        ps=new PrintStream(s.getOutputStream());
        ps.println(String.valueOf(area));
        ps.close();
        br.close();
        s.close();
        ss.close();
    }
}

```

Client side program:

```

import java.io.*;
import java.net.*;
class client
{
    public static void main(String[] args) throws Exception
    {
        Socket s=new Socket("localhost",8080);
        BufferedReader br;
        PrintStream ps;
        String str;

        System.out.println("enter the radius to send the server:");
        br=new BufferedReader(new InputStreamReader(System.in));
        ps=new PrintStream(s.getOutputStream());
        ps.println(br.readLine());

        br=new BufferedReader(new
InputStreamReader(s.getInputStream()));
        str=br.readLine();
        System.out.println("area of the circle:"+str);
        br.close();
        ps.close();
    }
}

```

Output::server side

C:\javac server.java

C:\java server

Wait for client Request

Received radius 20

Client side

C:\javac client.java

C:\java client

Enter the radius to send the server

20

Area of the circle is::1256.0

21. Write a java program that illustrate how run time polymorphism is achieved

```
class figure
{
    double dim1,dim2;
    figure(double x,double y)
    {
        dim1=x;
        dim2=y;
    }
    double area()
    {
        System.out.println("area undefiend");
        return 0;
    }
}
class rectangle extends figure
{
    rectangle(double a,double b)
    {
        super(a,b);
    }
    double area()
    {
        System.out.println("rectangle area");
        return dim1*dim2;
    }
}
```

```

}
class triangle extends figure
{
    triangle(double x,double y)
    {
        super(x,y);
    }
    double area()
    {
        System.out.println("triangle area");
        return (dim1*dim2)/2;
    }
}
class run
{
    public static void main(String args[])
    {
        figure obj=new figure(10,10);
        rectangle obj1=new rectangle(9,5);
        triangle obj2=new triangle(10,8);
        figure a;
        a=obj1;
        System.out.println("area="+a.area());
        a=obj2;
        System.out.println("area="+a.area());
        a=obj;
        System.out.println("area="+a.area());
    }
}

```

OUTPUT

C:\502>javac run.java

```

C:\502>java run
rectangle area
area=45.0
triangle area
area=40.0
area undefined
area=0.0

```

