

CP - VI – PROGRAMMING IN JAVA LAB

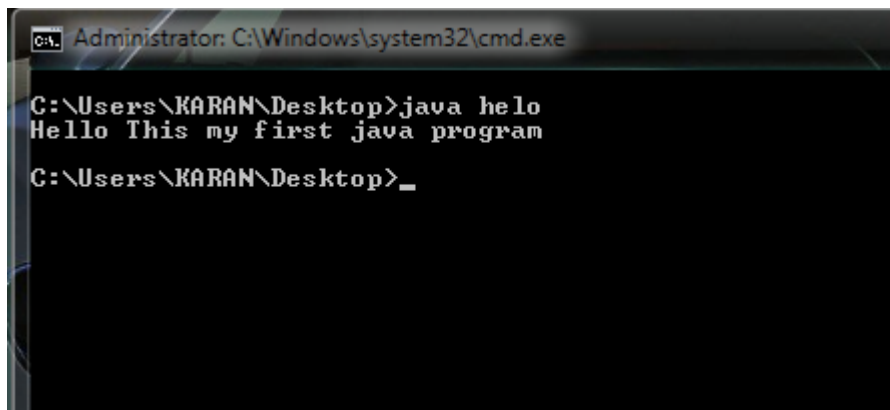
1. Classes and Objects
2. Control Statements
3. Constructors
4. Method Overloading and Overriding
5. String Handling
6. Inheritance
7. Packages
8. Interfaces
9. Exception Handling
10. Threads
11. File Processing
12. Graphics Methods
13. AWT controls
14. AWT Event Handling

PROGRAMMING IN JAVA LAB - SAMPLE PROGRAMS

1. Write a program to print sample program

coding

```
public class helo
{
public static void main(String args[])
{
System.out.println("Hello This my first java program");
}
}
```

A screenshot of a Windows command prompt window. The title bar reads "Administrator: C:\Windows\system32\cmd.exe". The command prompt shows the following text:

```
C:\Users\KARAN\Desktop>java helo
Hello This my first java program
C:\Users\KARAN\Desktop>_
```

2. Write a program to print fibonacci series upto 10 numbers

coding

```
public class Fibonacci
{
    public static void main(String[] args)
    {
        int limit = 10;
        long[] series = new long[limit];
        series[0] = 0;
        series[1] = 1;
```

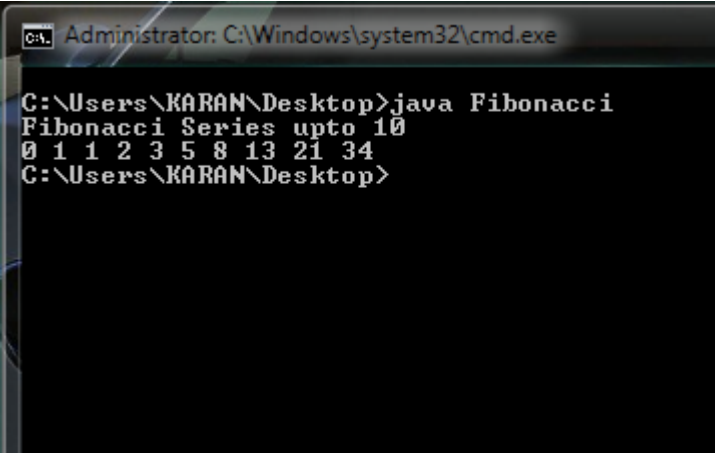
```

for(int i=2; i < limit; i++)
{
    series[i] = series[i-1] + series[i-2];
}

System.out.println("Fibonacci Series upto " + limit);

for(int i=0; i< limit; i++)
{
    System.out.print(series[i] + " ");
}
}
}

```



```

Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Desktop>java Fibonacci
Fibonacci Series upto 10
0 1 1 2 3 5 8 13 21 34
C:\Users\KARAN\Desktop>

```

- 3. WRITE A PROGRAM TO PERFORM MATHEMATICAL OPERATIONS. CREATE A CLASS CALLED ADDSUB WITH METHODS TO ADD AND SUBTRACT. CREATE ANOTHER CLASS CALLED MULTDIV THAT EXTENDS FROM ADDSUB CLASS TO USE THE MEMBER DATA OF THE SUPERCLASS. MULTDIV SHOULD HAVE METHODS TO MULTIPLY AND DIVIDE. A MAIN METHOD SHOULD ACCESS THE METHOD AND PERFORM THE MATHEMATICAL OPERATIONS.**

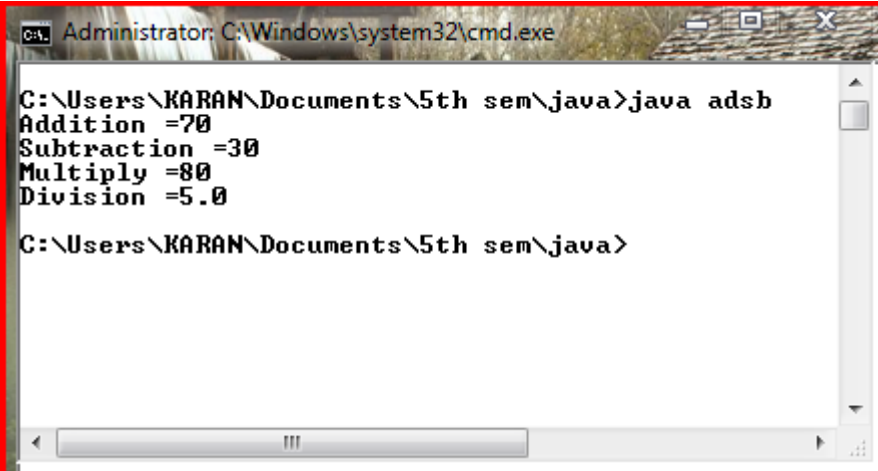
CODING

```

class addsub
{
    int num1;
    int num2;
    addsub(int n1, int n2)
    {
        num1 = n1;
        num2 = n2;
    }
    int add()
    {
        return num1+num2;
    }
    int sub()
    {
        return num1-num2;
    }
}

```

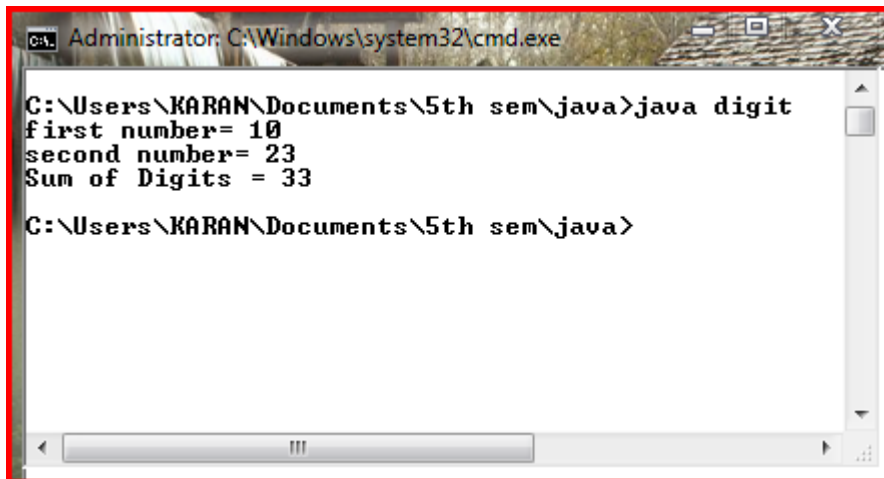
```
}  
class multdiv extends addsub  
{    public multdiv(int n1, int n2)  
    {  
        super(n1, n2);  
    }  
    int mul()  
    {return num1*num2;}  
    float div()  
    {return num2/num1;}  
    public void display()  
    {  
        System.out.println("Number 1 :" + num1);  
        System.out.println("Number 2 :" + num2);  
    }  
}  
  
    public class adsb  
{  
    public static void main(String arg[])  
    {  
        addsub r1=new addsub(50,20);  
        int ad = r1.add();  
        int sb = r1.sub();  
        System.out.println("Addition =" +ad);  
        System.out.println("Subtraction =" +sb);  
        multdiv r2 =new multdiv(4,20);  
        int ml = r2.mul();  
        float dv =r2.div();  
        System.out.println("Multiply =" +ml);  
        System.out.println("Division =" +dv);}}
```



```
Administrator: C:\Windows\system32\cmd.exe  
C:\Users\KARAN\Documents\5th sem\java>java adsb  
Addition =70  
Subtraction =30  
Multiply =80  
Division =5.0  
C:\Users\KARAN\Documents\5th sem\java>
```

4. WRITE SIMPLE PROGRAM TO CALCULATE THE SUM OF DIGITS OF ANY NUMBER.**Coding**

```
class digit
{
    public static void main(String args[])
    {
        int sum;
        int num1 = 10;
        int num2 = 23;
        {
            System.out.println("first number= " +num1);
            System.out.println("second number= " +num2);
            sum=num1+num2;
            System.out.println("Sum of Digits = " +sum);
        }
    }
}
```



```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Documents\5th sem\java>java digit
first number= 10
second number= 23
Sum of Digits = 33
C:\Users\KARAN\Documents\5th sem\java>
```

5. DESIGN A CLASS TO REPRESENT A BANK ACCOUNT. INCLUDE THE FOLLOWING MEMBERS:**DATA MEMBERS:****NAME OF THE DEPOSITOR****ACCOUNT NUMBER****TYPE OF ACCOUNT****BALANCE AMOUNT IN THE ACCOUNT****METHODS:****TO ASSIGN INITIAL VALUES****TO DEPOSIT AN AMOUNT****TO WITHDRAW AN AMOUNT AFTER CHECKING BALANCE****TO DISPLAY THE NAME AND BALANCE**

Coding

```

class account
{
    String name;
    int acc_no;
    String type;
    int bal;
    int amount;

    account()
    {
        name="karandeep";
        acc_no=2905;
        type="savings";
        bal=6000;
    }
    int withdraw()
    {
        amount=1000;
        bal-=amount;
        return bal;
    }
    int deposit()
    {
        amount=500;
        bal+=amount;
        return bal;
    }

    void display()
    {
        System.out.println(name + " Your A/c no :" +acc_no + " Your A/c type :"
+type);

        System.out.print("Your Balance is Rs." + bal);

        withdraw();

        System.out.print("\nYour Balance after withdrawl is Rs" + bal);

        deposit();

        System.out.print("\nYour Balance after deposit is Rs." + bal);
    }
}

public class karan
{
    public static void main(String args[])
    {
        account b1=new account();

        b1.display();
    }
}

```

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Documents\5th sem\java>java karan
karandeep Your A/c no :2905 Your A/c type :savings
Your Balance is Rs.6000
Your Balance after withdrawl is Rs5000
Your Balance after deposit is Rs.5500
C:\Users\KARAN\Documents\5th sem\java>
```

6. Write a program TO DISPLAY AREA OF RECTANGLE AND TRIANGLE USING INHERITANCE

Coding

```
class Shape
{
    int dim1;
    int dim2;
    Shape(int x , int y)
    {
        dim1=x;
        dim2=y;
    }

    void display()
    {
        System.out.println(dim1);
        System.out.println(dim2);
    }
}
class rectangle extends Shape
{
    rectangle(int x, int y)
    {
        super(x, y);
    }
    int area()
    {
        return dim1*dim2;
    }

    void display()
    { System.out.println("In rectangle class");
      super.display();
    }
}
```

```

class triangle extends Shape
{
    triangle(int x, int y)
    {
        super(x, y);
    }
    float area()
    {
        return (0.5f)*dim1*dim2;
    }
}
class sh
{
    public static void main(String arg[])
    {
        rectangle r1=new rectangle(20,20);
        int ar=r1.area();
        System.out.println("Area of Rectangle " +ar);
        r1.display();
        triangle t1=new triangle(20,25);
        float ar1=t1.area();
        System.out.println("Area of Triangle " +ar1);
    }
}

```

```

Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Documents\5th sem\java>java sh
Area of Rectangle 400
In rectangle class
20
20
Area of Triangle 250.0
C:\Users\KARAN\Documents\5th sem\java>

```

7. WRITE A SIMPLE PROGRAM TO DISPLAY A "*" I TRIANGLE SHAPE. OUTPUT WILL BE LIKE THIS

```

*

* * *

* * * * *

```

coding

```

class star
{
    public static void main (String args[])

```

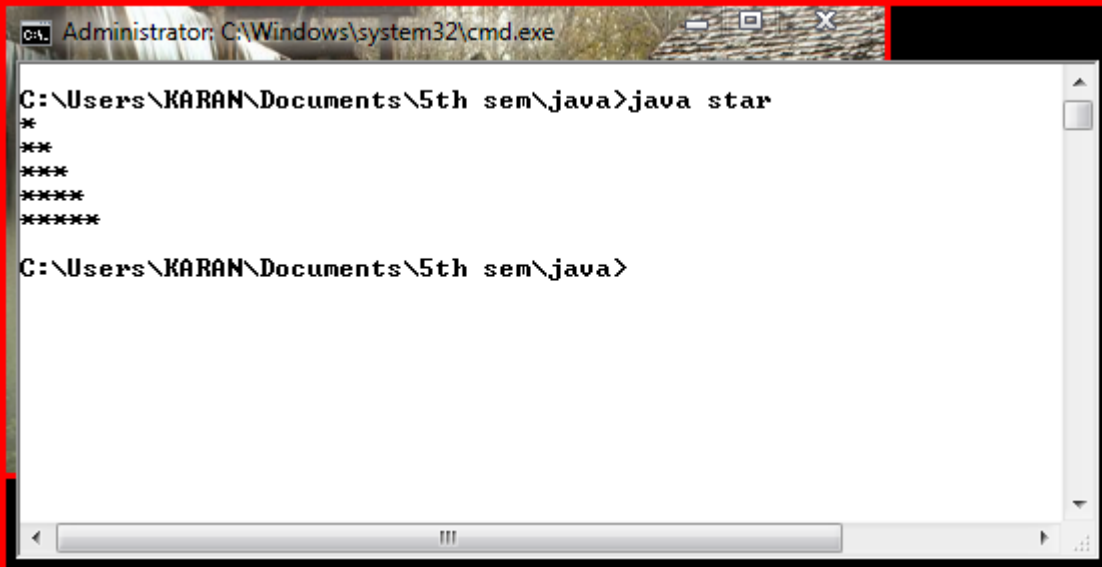


```

{
int i;
int j;

for ( i=1; i<=5; i++)
{
    for (j=1; j<=i; j++)
    {
        System.out.print("*");
    }
    System.out.println();
}
}
}

```



```

Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Documents\5th sem\java>java star
*
**
***
****
*****
C:\Users\KARAN\Documents\5th sem\java>

```

8. CREATE A CLASS WITH A DEFAULT CONSTRUCTOR (ONE THAT TAKES NO ARGUMENTS) THAT PRINTS A MESSAGE. CREATE AN OBJECT OF THIS CLASS.

Coding

```
class rec
```

```
{
```

```
    int a;
    int b;
```

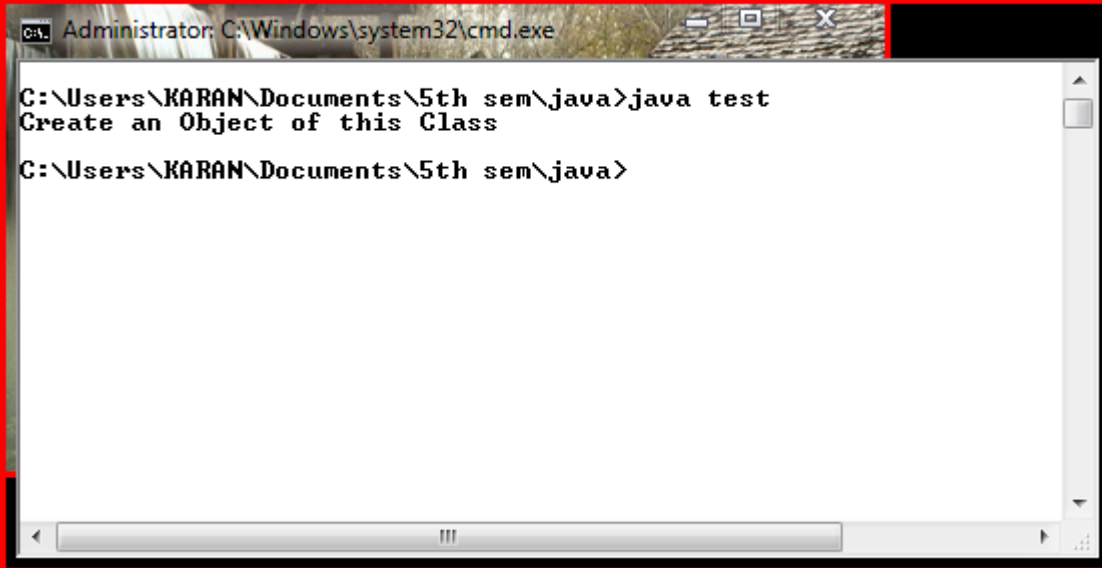
```
    rec ( )
```

```
    {
        a=5;
        b=10;
    }
```

```
        int area()
```

```
        {
            return a*b;
        }
```

```
    }  
}  
  
class test  
{  
    public static void main (String args[ ])  
    {  
        System.out.println("Create an Object of this Class");  
    }  
}
```



```
Administrator: C:\Windows\system32\cmd.exe  
C:\Users\KARAN\Documents\5th sem\java>java test  
Create an Object of this Class  
C:\Users\KARAN\Documents\5th sem\java>
```

9. Write a program TO STORE NUMBERS IN ARRAY

```
class testarra  
  
{  
    public static void main(String args[])  
    {  
        int anarray[];  
        anarray = new int[10];  
        anarray[1] = 90;  
        anarray[2] = 80;  
        anarray[3] = 70;  
        anarray[4] = 50;  
        anarray[5] = 60;  
  
        System.out.println(anarray[1]);  
        System.out.println(anarray[4]);  
        System.out.println(anarray[2]);  
    }  
}
```

```
}
}
```

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Documents\5th sem\java>java testarra
90
50
80
C:\Users\KARAN\Documents\5th sem\java>
```

10. Write a program TO STORE STRING IN ARRAY

Coding

Class testarraystr

```
{
    public static void main(String args[])
    {
        String anarray[];
        anarray = new String[10];
        anarray[1] = "Karandeep";
        anarray[2] = "kandy";
        anarray[3] = "Mandy";
        anarray[4] = "Sahil";
        anarray[5] = "Dinesh";

        System.out.println(anarray[1]);
        System.out.println(anarray[4]);
        System.out.println(anarray[2]);
    }
}
```

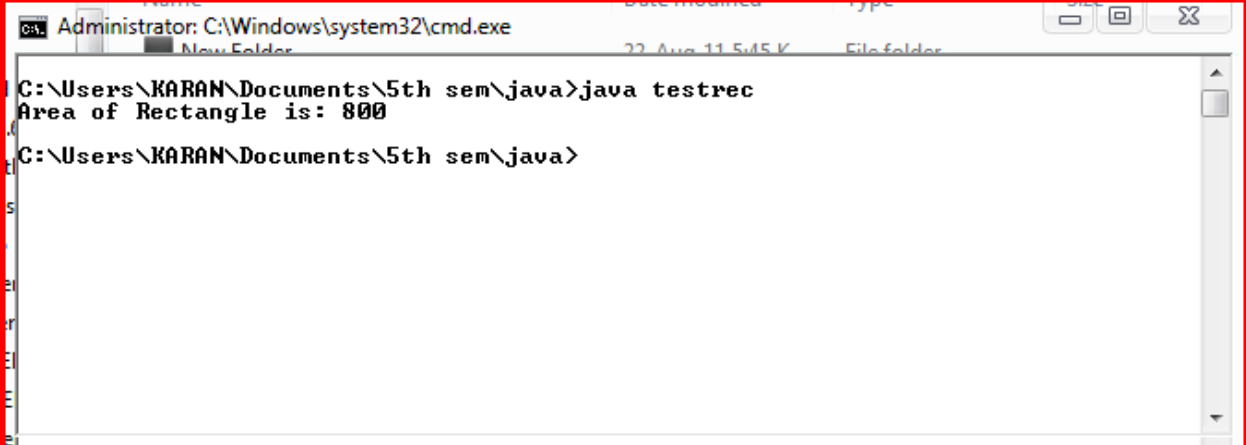
```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Documents\5th sem\java>java testarraystr
Karandeep
Sahil
kandy
C:\Users\KARAN\Documents\5th sem\java>
```

11. Write a program TO PRINT AREA OF RECTANGLE USING FUNCTION**Coding**

```
class rectangle
{
    int length;
    int width;

    int area()
    {
        return length*width;
    }
}

public class testrec
{
    public static void main(String args[])
    {
        rectangle r1 = new rectangle();
        r1.length=20;
        r1.width=40;
        int r1area = r1.area();
        System.out.println("Area of Rectangle is: " + r1area);
    }
}
```



```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Documents\5th sem\java>java testrec
Area of Rectangle is: 800
C:\Users\KARAN\Documents\5th sem\java>
```

12. WRITE A JAVA PROGRAM TO FIND THE VOLUME OF A SPHERE AND A CONE.**Coding**

```
class Shape
{
    int radius;
    int height;
    Shape(int x , int y)
    {
        radius=x;
        height=y;
    }

    void display()
    {
        System.out.println(radius);
        System.out.println(height);
    }
}

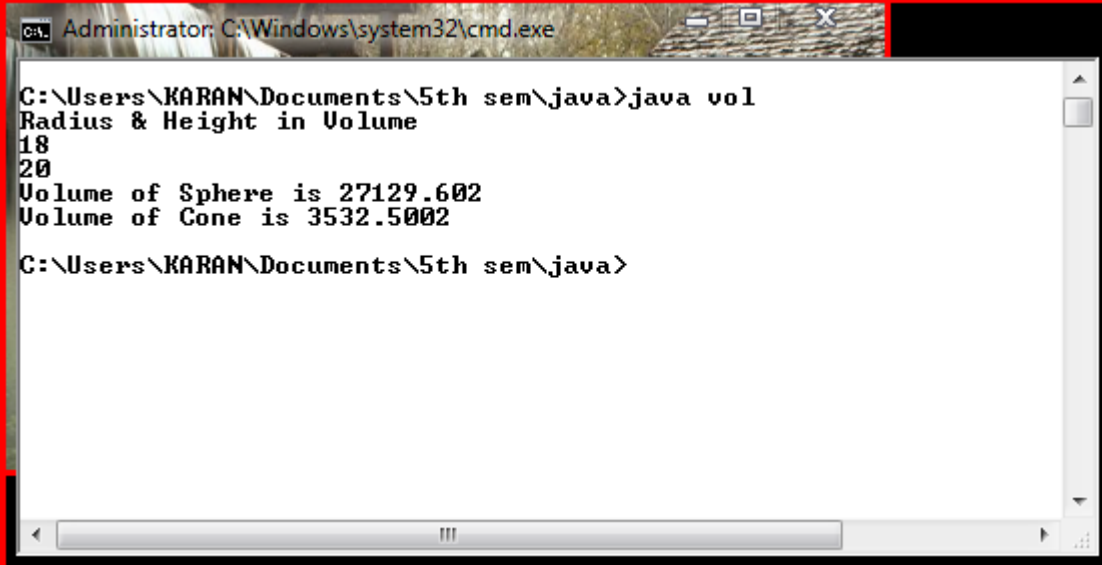
class sphere extends Shape
{
    sphere(int x, int y)
    {
        super(x, y);
    }
    float volume()
    {
        return (4/3f)*(3.14f)*radius*radius*height;
    }

    void display()
    { System.out.println("Radius & Height in Volume");
      super.display();
    }
}

class cone extends Shape
{
    cone(int x, int y)
    {
        super(x, y);
    }
    float volume()
    {
        return (1/3f)*(3.14f)*radius*radius*radius;
    }
}

class vol
{
    public static void main(String arg[])
    {
        sphere r1=new sphere(18,20);
        float ar=r1.volume();
        r1.display();
        System.out.println("Volume of Sphere is " +ar);
        cone t1=new cone(15,15);
        float ar1=t1.volume();
    }
}
```

```
        System.out.println("Volume of Cone is " +ar1);  
    }  
}
```



```
Administrator: C:\Windows\system32\cmd.exe  
C:\Users\KARAN\Documents\5th sem\java>java vol  
Radius & Height in Volume  
18  
20  
Volume of Sphere is 27129.602  
Volume of Cone is 3532.5002  
C:\Users\KARAN\Documents\5th sem\java>
```

13. WRITE A JAVA PROGRAM THAT TAKES A STRING AND CONVERTS IT INTO UPPERCASE AND LOWERCASE LETTERS

Coding

```
public class prog  
{  
    public static void main(String args[])  
    {  
        String s = "KaRaN, SaGGu!";  
  
        String supper = s.toUpperCase();  
        String slower = s.toLowerCase();  
  
        System.out.println("The Upper case is"+supper);  
        System.out.println("The Lower case is"+slower);  
    }  
}
```

```

}
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\KARAN>cd desktop
C:\Users\KARAN\Desktop>java prog
The Upper case isKARAN, SAGGU!
The Lower case iskaran, saggu!
C:\Users\KARAN\Desktop>

```

14. WRITE A JAVA PROGRAM TO CONVERT RUPEES TO DOLLARS.

Coding

```

class display
{
    public static void main(String args[])
    {
        currency ob= new currency();
        double money=ob.convert(25.0);
        System.out.println("The converted money in dollar is = "+money);
    }
}
class currency
{
    double convert(double rupee)
    {
        double doll=45*rupee;
        return doll;
    }
}

```

```

Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Desktop>java display
The money in dollar is = 1125.0
C:\Users\KARAN\Desktop>

```

```
class pwr
{
    double r;

    double change(int x, int y)
    {
        r = Math.pow(x,y);
        return r;
    }

    double change(int x, float y)
    {
        r = Math.pow(x,y);
        return r;
    }

    double change(float x, float y)
    {
        r = Math.pow(x,y);
        return r;
    }
}

public class pw
{
    public static void main(String args[])
    {
        pwr p = new pwr();

        double a = p.change(2,2);

        double b = p.change(2,3.2F);

        double c = p.change(4.3F,3.6F);

        System.out.println("Integer combination: "+a);

        System.out.println("Integer - float combination: "+b);

        System.out.println("Float combination: "+c);
    }
}
```


A screenshot of a Windows command prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The window shows the execution of two Java programs. The first program, "display", outputs "The money in dollar is = 1125.0". The second program, "pw", outputs three lines: "Integer combination: 4.0", "Integer - float combination: 9.189587143708977", and "Float combination: 190.7596201722023". The prompt is currently at "C:\Users\KARAN\Desktop>".

```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Desktop>java display
The money in dollar is = 1125.0
C:\Users\KARAN\Desktop>java pw
Integer combination: 4.0
Integer - float combination: 9.189587143708977
Float combination: 190.7596201722023
C:\Users\KARAN\Desktop>
```

abstract class shape

```
{  int dim1;

    int dim2;

    shape(int x, int y)

    {    dim1 = x;

        dim2 = y;

    }

    abstract double area();

}
```

class rectangle extends shape

```
{  rectangle(int x, int y)

    {  super(x,y);

    }

    double area()

    {  return dim1*dim2;

    }

}
```

```
class triangle extends shape
```

```
{  triangle(int x, int y)

    {  super(x,y);

    }
    double area()

    {  return (0.5F)*dim1*dim2;

    }
}
```

```
public class abshape
```

```
{  public static void main(String args[])

    {  rectangle r1 = new rectangle(10,20);

        triangle t1 = new triangle(20,15);

        double r = r1.area();

        double t = t1.area();

        System.out.println("Area of rectangle: " +r);

        System.out.println("Area of triangle: " +t);

    }

}
```



```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Desktop>java abshape
Area of rectangle: 200.0
Area of triangle: 150.0
C:\Users\KARAN\Desktop>
```

15. WRITE A JAVA PROGRAM THAT ACCEPTS TWO STRINGS AS COMMAND LINE ARGUMENTS. IT CHECKS FOR THE NUMBER OF COMMAND LINE ARGUMENTS. IF THEY ARE LESS OR MORE IT THROWS AN EXCEPTION GIVING AN APPROPRIATE MESSAGE.

CODING

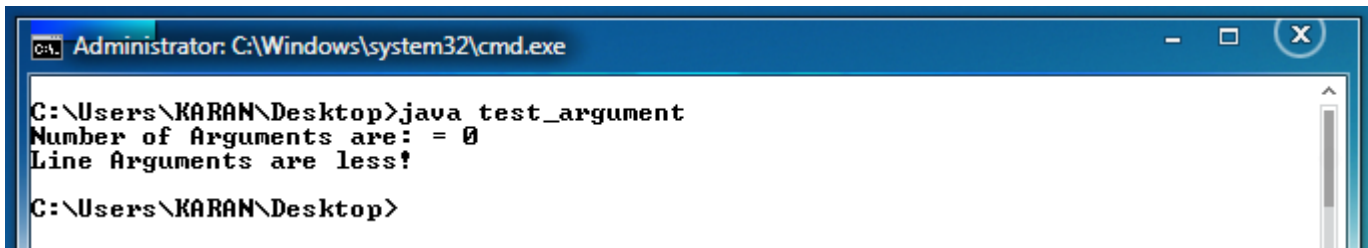
```
import java.lang.Exception;
class Exp extends Exception
{
    Exp(String msg)
    {
        super(msg);
    }
}

class test_argument
{
    public static void main(String args[])
    {

        String string;

        try
        {
            int count, i=0;

            count=args.length;
            System.out.println("Number of Arguments are: = " +count);
            if(count<2)
            {
                Exp ex=new Exp("Line Arguments are less!");
                throw ex;
            }
            else if(count>2)
            {
                Exp ex=new Exp("Line Arguments are more!");
                throw ex;
            }
            while(i<count)
            {
                string = args[i];
                i=i+1;
                System.out.println(i + ":" + "Number of Entered Arguments: " + string+ "!");
            }
        }
        catch(Exp ex)
        {
            System.out.println(ex.getMessage());
        }
    }
}
```



```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Desktop>java test_argument
Number of Arguments are: = 0
Line Arguments are less!
C:\Users\KARAN\Desktop>
```

16. WRITE APPLETS TO DRAW THE FOLLOWING SHAPES:

(I) CONE

(II) CYLINDER

(III) CUBE

(IV) SQUARE INSIDE A CIRCLE

(V) CIRCLE INSIDE A SQUARE

CODING

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

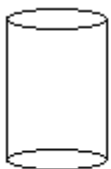
public class Shapes extends Applet
{
    public void paint(Graphics g)
    {
        /*Cylinder*/
        g.drawString("(a).Cylinder",10,110);
        g.drawOval(10,10,50,10);
        g.drawOval(10,80,50,10);
        g.drawLine(10,15,10,85);
    }
}
```

```

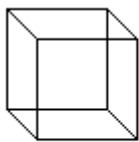
g.drawLine(60,15,60,85);
/*Cube*/
g.drawString("(b).Cube",95,110);
g.drawRect(80,10,50,50);
g.drawRect(95,25,50,50);
g.drawLine(80,10,95,25);
g.drawLine(130,10,145,25);
g.drawLine(80,60,95,75);
g.drawLine(130,60,145,75);
/*Squar Inside A Circle*/
g.drawString("(c).Squar Inside A Circle",150,110);
g.drawOval(180,10,80,80);
g.drawRect(192,22,55,55);
/*Circle Inside a Squar*/
g.drawString("(d).Circle Inside a Squar",290,110);
g.drawRect(290,10,80,80);
g.drawOval(290,10,80,80);

}
}

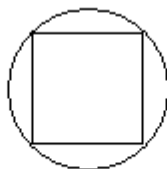
```



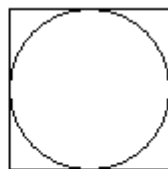
(a).Cylinder



(b).Cube



(c).Squar Inside A Circle



(d).Circle Inside a Squar

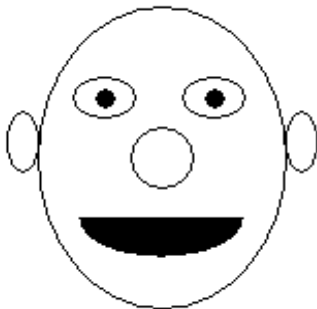
17. WRITE AN APPLET TO DISPLAY A FACE.**CODING**

```

import java.awt.*;
import java.applet.*;

public class face extends Applet
{
    public void paint (Graphics g)
    {
        g.drawOval (83, 10, 40, 30) ;
        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) ;
        g.fillArc (60, 125, 80, 40, 180, 180) ;
        g.drawOval (25, 92, 15, 30) ;
        g.drawOval (160, 92, 15, 30) ;
    }
}

```

**18. WRITE AN APPLET TO DISPLAY FIVE BUTTONS.****CODING**

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class myApplet extends Applet implements ActionListener{
    TextField t1,t2,t3;
    Label l1,l2,l3;
    Button addBtn,subBtn,multBtn,divBtn,percBtn;

    public void init(){

```

```

        t1=new TextField();
        t2=new TextField();
t3=new TextField();
l1=new Label("Num 1");
l2=new Label("Num 2");
l3=new Label("Result");
addBtn=new Button("+");
subBtn=new Button("-");
multBtn=new Button("*");
divBtn=new Button("/");
percBtn=new Button("%");
this.add(l1);
this.add(t1);
this.add(l2);
this.add(t2);
this.add(l3);
this.add(t3);
this.add(addBtn);
this.add(subBtn);
this.add(multBtn);
this.add(divBtn);
this.add(percBtn);
addBtn.addActionListener(this);
subBtn.addActionListener(this);
multBtn.addActionListener(this);
divBtn.addActionListener(this);
percBtn.addActionListener(this);
}

public void actionPerformed(ActionEvent ae)
{
String args=ae.getActionCommand();
if(args.equals("+")){
    int n1 = Integer.parseInt(t1.getText());
    int n2 = Integer.parseInt(t2.getText());
    int sum = n1+n2;
    t3.setText(Integer.toString(sum));
}
else if(args.equals("-")){
    int n1 = Integer.parseInt(t1.getText());
    int n2 = Integer.parseInt(t2.getText());
    int sub=n1-n2;
    t3.setText(Integer.toString(sub));
}
else if(args.equals("*")){
    int n1 = Integer.parseInt(t1.getText());
    int n2 = Integer.parseInt(t2.getText());
    int mult=n1*n2;
    t3.setText(Integer.toString(mult));
}
else if(args.equals("/")){

```

```

        int n1 = Integer.parseInt(t1.getText());
        int n2 = Integer.parseInt(t2.getText());
        double div=n1/n2;
        t3.setText(Double.toString(div));
    }
    else if(args.equals("%")){
        int n1 = Integer.parseInt(t1.getText());
        int n2 = Integer.parseInt(t2.getText());
        double perc=n1*n2/100;
        t3.setText(Double.toString(perc));
    }
    repaint();
}
}

```

Num 1 Num 2 Result

19. WRITE AN APPLLET TO ILLUSTRATE BORDERLAYOUT.

CODING

```

import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class BorderLayoutApplet extends Applet implements ActionListener{
    Button green=new Button("green");
    Button red=new Button("red");
    Button blue=new Button("blue");
    Button yellow=new Button("yellow");
    String msg="BorderLayoutDemo";
    TextArea area;

    public void init()
    {
        setLayout(new BorderLayout());
        area = new TextArea();
        area.setText(msg);
        add(green, BorderLayout.NORTH);
        add(red, BorderLayout.SOUTH);
        add(blue, BorderLayout.EAST);
        add(yellow, BorderLayout.WEST);
        add(area, BorderLayout.CENTER);
        green.addActionListener(this);
        red.addActionListener(this);
        blue.addActionListener(this);
        yellow.addActionListener(this);
    }
}

```



```
public void actionPerformed (ActionEvent a)
{
String str= a.getActionCommand();
if(str.equals("green"))
{
Font myfont = new Font("Garamond",Font.BOLD,25);

area.setFont(myfont);
area.setForeground(Color.green);
}
else if(str.equals("red"))
{
area.setForeground(Color.red);
}
else if(str.equals("blue"))
{
area.setForeground(Color.blue);
}
else if(str.equals("yellow"))
{
area.setForeground(Color.yellow);
}

else
{
Font myfont = new Font("Garamond",Font.ITALIC,25);
area.setFont(myfont);
area.setForeground(Color.cyan);
}
repaint();
}

public void paint(Graphics g)
{}
}
```

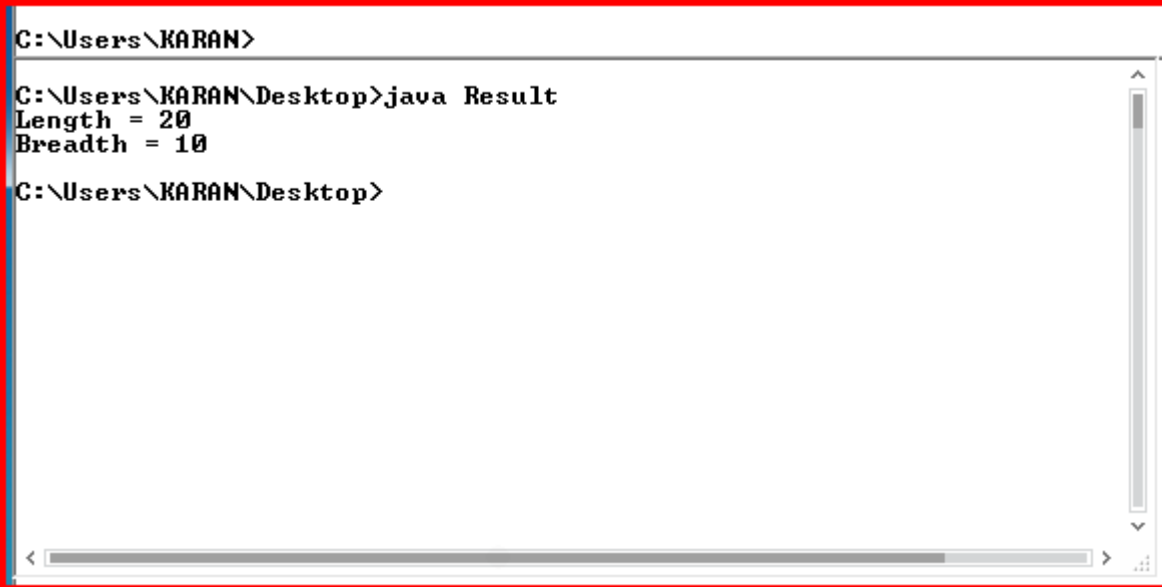


20. DESIGN AND WRITE A JAVA PROGRAM TO DEFINE A CLASS CALLED RECTANGLE THAT CONTAINS MEMBERS FOR REPRESENTING ITS LENGTH AND BREADTH. PROVIDE MEMBERS TO GET AND SET THESE ATTRIBUTES.

CODING

```
class Rectangle
{
int length, breadth;
public void getlen()
{
length = 20;
}
public int setlen()
{
return length;
}
public void getbre()
{
breadth = 10;
}
public int setbre()
{
return breadth;
}
}
public class Result
{
```

```
public static void main(String args[])  
{  
    Rectangle r = new Rectangle();  
    r.getlen();  
    int l = r.setlen();  
    r.getbre();  
    int b = r.setbre();  
    System.out.println("Length = " + l);  
    System.out.println("Breadth = " + b);  
}  
}
```



```
C:\Users\KARAN>  
C:\Users\KARAN\Desktop>java Result  
Length = 20  
Breadth = 10  
C:\Users\KARAN\Desktop>
```

21. CREATE AN ABSTRACT CLASS CALLED FIGURE THAT HAS AN ABSTRACT METHOD CALLED DRAW (). MAKE THE SUBCLASSES CALLED FILLED_RECTANGLE, FILLED_ARC AND OVERRIDE THE DRAW METHOD IN WHICH YOU WOULD PRINT THE MESSAGE REGARDING THE CURRENT OBJECT.

CODING

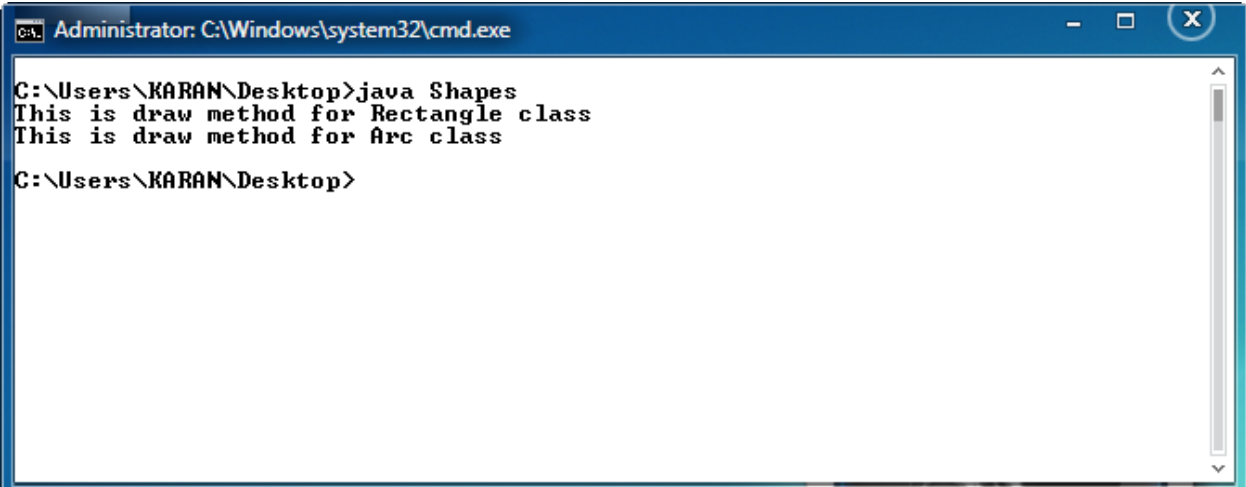
```
abstract class Figure
{
    abstract void draw();
}

class Filled_Rectangle extends Figure
{
    public void draw()
    {
        System.out.println("This is draw method for Rectangle class");
    }
}

class Filled_Arc extends Figure
{
    public void draw()
    {
        System.out.println("This is draw method for Arc class");
    }
}

public class Shapes
```

```
{  
  
    public static void main(String args[])  
  
    {  
  
        Filled_Rectangle r = new Filled_Rectangle();  
  
        r.draw();  
  
        Filled_Arc a = new Filled_Arc();  
  
        a.draw();  
  
    }  
  
}
```



The screenshot shows a Windows command prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The prompt is at "C:\Users\KARAN\Desktop>". The user has entered the command "java Shapes", and the output is "This is draw method for Rectangle class" followed by "This is draw method for Arc class". The prompt is now "C:\Users\KARAN\Desktop>".

```
Administrator: C:\Windows\system32\cmd.exe  
C:\Users\KARAN\Desktop>java Shapes  
This is draw method for Rectangle class  
This is draw method for Arc class  
C:\Users\KARAN\Desktop>
```

**22.WRITE A JAVA PROGRAM TO CREATE 5 THREADS BY EXTENDING
THREAD CLASS.****CODING**

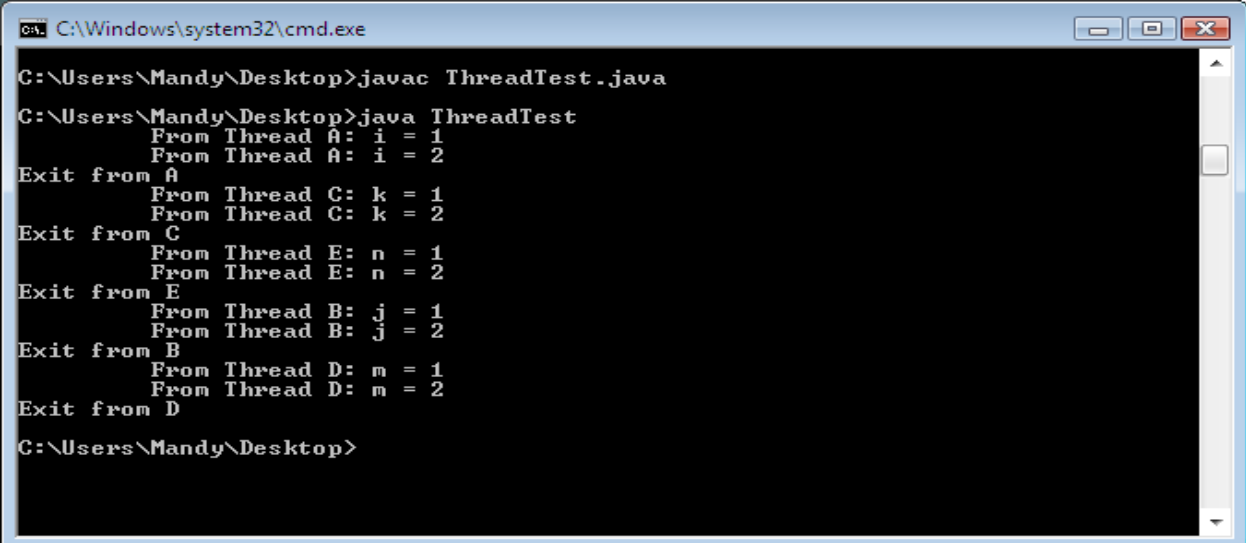
```
class A extends Thread
{
public void run()
{
for(int i=1; i<=2; i++)
{
System.out.println("\t From Thread A: i = " +i);
}
System.out.println("Exit from A");
}
}

class B extends Thread
{
public void run()
{
for(int j=1; j<=2; j++)
{
System.out.println("\t From Thread B: j = " +j);
}
System.out.println("Exit from B");
}
}

class C extends Thread
```

```
{  
  
public void run()  
  
{  
  
for(int k=1; k<=2; k++)  
  
{  
  
System.out.println("\t From Thread C: k = " +k);  
  
}  
  
System.out.println("Exit from C");  
  
}  
  
}  
  
class D extends Thread  
  
{  
  
public void run()  
  
{  
  
for(int m=1; m<=2; m++)  
  
{  
  
System.out.println("\t From Thread D: m = " +m);  
  
}  
  
System.out.println("Exit from D");  
  
}  
  
}  
  
class E extends Thread  
  
{  
  
public void run()  
  
{  
  
for(int n=1; n<=2; n++)
```

```
{  
    System.out.println("\t From Thread E: n = " +n);  
}  
  
System.out.println("Exit from E");  
  
}  
  
}  
  
public class ThreadTest  
{  
  
    public static void main(String args[])  
    {  
  
        new A().start();  
  
        new B().start();  
  
        new C().start();  
  
        new D().start();  
  
        new E().start();  
  
    }  
  
}
```



```
C:\Windows\system32\cmd.exe  
C:\Users\Mandy\Desktop>javac ThreadTest.java  
C:\Users\Mandy\Desktop>java ThreadTest  
    From Thread A: i = 1  
    From Thread A: i = 2  
Exit from A  
    From Thread C: k = 1  
    From Thread C: k = 2  
Exit from C  
    From Thread E: n = 1  
    From Thread E: n = 2  
Exit from E  
    From Thread B: j = 1  
    From Thread B: j = 2  
Exit from B  
    From Thread D: m = 1  
    From Thread D: m = 2  
Exit from D  
C:\Users\Mandy\Desktop>
```


23.WRITE A JAVA PROGRAM TO CREATE 5 THREADS BY IMPLEMENTING RUNNABLE INTERFACE.**CODING**

```
class MyRunnable implements Runnable{

private int a;

public MyRunnable(int a){

this.a = a;

}

public void run(){

for (int i = 1; i <= a; ++i){

System.out.println(Thread.currentThread().getName() + " is " + i);

try{

Thread.sleep(1000);

}

catch (InterruptedException e){}

}

}

}

class MainMyThread{

public static void main(String args[]){

MyRunnable thr1, thr2;

thr1 = new MyRunnable(5);

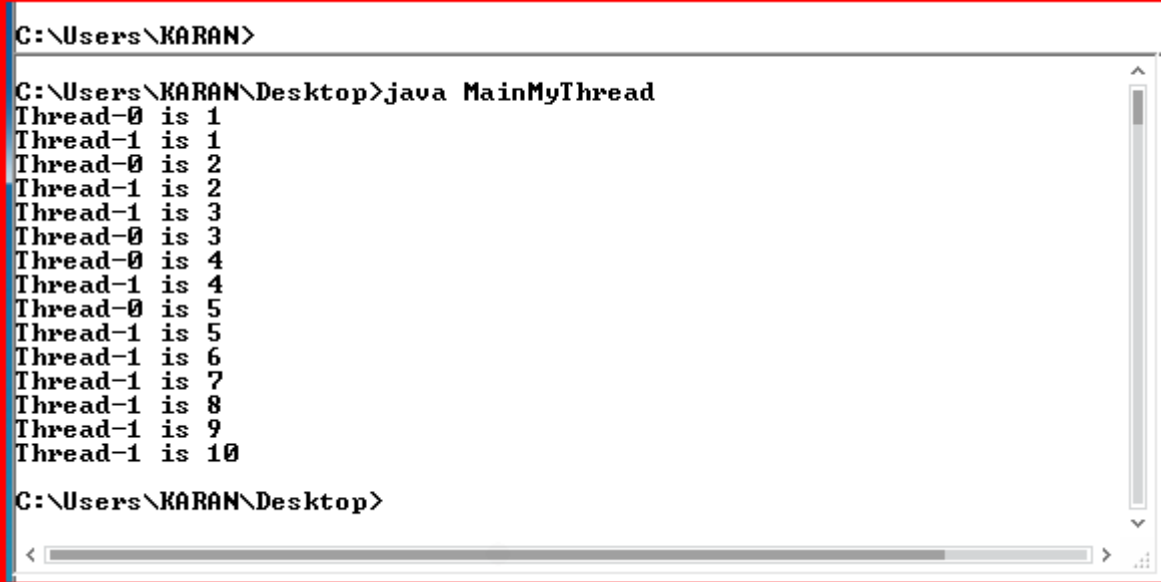
thr2 = new MyRunnable(10);

Thread t1 = new Thread(thr1);

Thread t2 = new Thread(thr2);

t1.start();
```

```
t2.start();
}
}
```



```
C:\Users\KARAN>
C:\Users\KARAN\Desktop>java MainMyThread
Thread-0 is 1
Thread-1 is 1
Thread-0 is 2
Thread-1 is 2
Thread-1 is 3
Thread-0 is 3
Thread-0 is 4
Thread-1 is 4
Thread-0 is 5
Thread-1 is 5
Thread-1 is 6
Thread-1 is 7
Thread-1 is 8
Thread-1 is 9
Thread-1 is 10
C:\Users\KARAN\Desktop>
```

24. WRITE A JAVA PROGRAM THAT HAS INTEGER VARIABLES A, B, C AND RESULT AS FLOAT. STORE SOME VALUES IN THEM AND APPLY THE FORMULA RESULT = A/(B-C). CATCH THE PROBABLE EXCEPTION.

CODING

```
public class ArEx
{
    public static void main(String args[])
    {
        try
        {
            int a = 20;
```

```
        int b = 4;

        int c = 4;

        float d;

        d = a/(b-c);

    }

    catch(ArithmeticException e)

    {

        System.out.println("Error Occurred: " +e.getMessage());

    }

}

}
```



```
Administrator: C:\Windows\system32\cmd.exe
C:\Users\KARAN\Desktop>javac ArEx.java
C:\Users\KARAN\Desktop>java ArEx
Error Occurred: / by zero
C:\Users\KARAN\Desktop>
```