

**Ex. No.1 :** Create a class Box. Define the **instance variables** width, height and depth of type **double**. Create a **Constructor** for the class Box. With the help of ‘**this keyword**’ assign the parameters. Define a method volume to return the Volume of the Box.

**Date :**

**Aim**

Write a java program to calculate volume of the box using ‘constructor’ and ‘this’ operator.

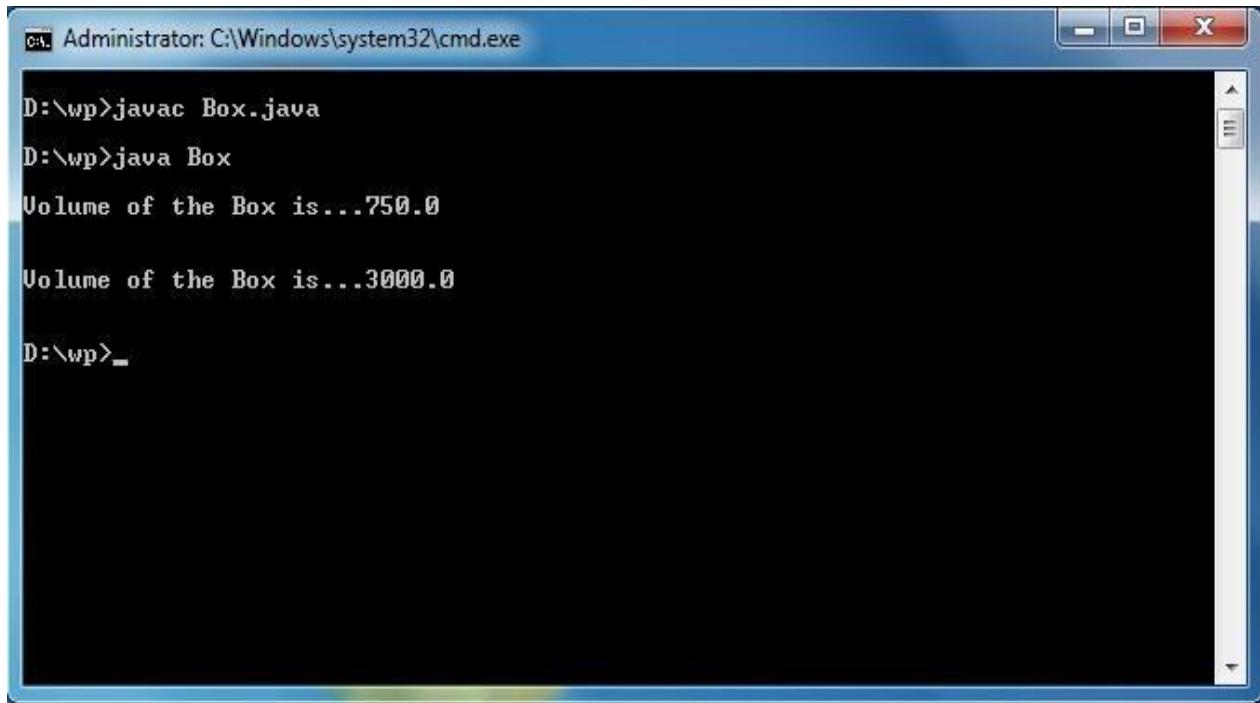
**Algorithm**

1. Start the program.
2. Create a class Box.
3. Create a constructor Box and method display.
4. Variables can initiate into the constructor().
4. Display the result.
5. Stop the program.

**Program :**

```
class Box
{
    double width, height, depth;
    Box(double w, double h, double dd)
    {
        this.width = w;
        this.height = h;
        this.depth = dd;
    }
    void display()
    {
        System.out.println("\n"+ "Volume of the Box is..." + (width*height*depth)+"\n");
    }
    public static void main(String arg[])
    {
        Box a,b;
        a = new Box(5,10,15);
        a.display();
        b = new Box(10,15,20);
        b.display();
    }
}
```

**Output :**



D:\wp>javac Box.java  
D:\wp>java Box  
Volume of the Box is...750.0  
Volume of the Box is...3000.0  
D:\wp>\_

A screenshot of a Windows Command Prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The window shows the execution of a Java program named "Box". The first command "javac Box.java" compiles the Java source code. The second command "java Box" runs the compiled class, outputting "Volume of the Box is...750.0" and "Volume of the Box is...3000.0" to the console. The window has a standard blue title bar and a black background.

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

**Ex. No. : 2.** Write a Java program to illustrate **Method Overloading** for the following: Create a main class overloading. Within this class get the necessary inputs and call the different forms of area methods to calculate the area of the following shapes.

- (i) Square    (ii) Rectangle    (iii) Circle

**Date:**

**Aim**

Write a java program to demonstrate Method Overloading concept.

**Algorithm**

1. Start the program.
2. Create a class Overload Demo.
3. Create a method name area() and pass values into the different argument.
4. Calculate the various shape area.
5. Display result by using object.
6. Stop the program.

## **Program**

```
class OverloadDemo

{

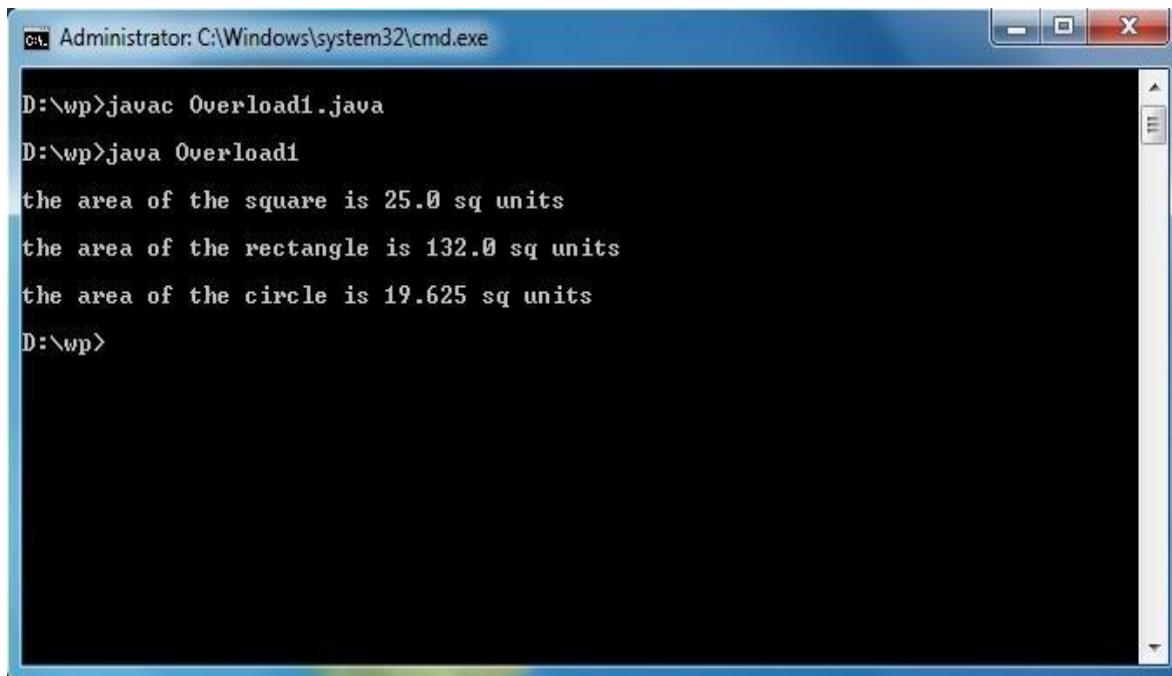
    void area(float x)
    {
        System.out.println("\nthe area of the square is "+Math.pow(x, 2)+" sq units");
    }

    void area(float x, float y)
    {
        System.out.println("\nthe area of the rectangle is "+x*y+" sq units");
    }

    void area(double x)
    {
        double z = 3.14 * x * x;
        System.out.println("\nthe area of the circle is "+z+" sq units");
    }
}

class Overload1
{
    public static void main(String args[])
    {
        OverloadDemo ob = new OverloadDemo();
        ob.area(5);
        ob.area(11,12);
        ob.area(2.5);
    }
}
```

**Output :**



D:\wp>javac Overload1.java  
D:\wp>java Overload1  
the area of the square is 25.0 sq units  
the area of the rectangle is 132.0 sq units  
the area of the circle is 19.625 sq units  
D:\wp>

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

**Ex. No. : 3.** Write a Java program to illustrate **Method Overriding** for the following: Create a main class overriding. Get the empid, name, grade, salary, designation from the user. Create a base class employee and display the employee details using show() method. Create a derived class sportsemployee and display the sports details using show() method.

**Date :**

**Aim**

Write a java program to demonstrate the Method Overriding concept.

**Algorithm**

1. Start the program.
2. Create the class override1.
3. Create the super class employee and derived class sportsemployee.
4. Create a method name show().
5. Display result by using object.
6. Stop the program.

## **Program**

```
class employee
{
    int empid;
    String name, designation;
    double salary;

    employee(int eid, String na, String des, double sa)
    {
        empid = eid;
        name = na;
        designation = des;
        salary =sa;
    }

    void show()
    {
        System.out.println("\nEmployee Id : "+empid);
        System.out.println("\nEmployee Name : "+name);
        System.out.println("\nEmployee Designation : "+designation);
        System.out.println("\nEmployee Salary : "+salary);
    }
}

class sportsemployee extends employee
{
    String grade;
    sportsemployee(int eid, String na, String des, String gra, double sa)
    {
        super(eid,na,des,sa);
        grade = gra;
    }
}
```

```
void show()
{
    System.out.println("\nSports Employee Grade : "+grade);
}

class override1
{
    public static void main(String arg[])
    {
        employee ee = new employee(101,"senthil","Professor",50000);
        sportemployee t = new
sportemployee(101,"senthil","Professor","Batsman",50000);

        employee em;
        em = ee;
        em.show();
        em = t;
        em.show();
    }
}
```

**Output :**



A screenshot of a Windows Command Prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The window shows the following text:

```
D:\wp>javac override1.java
D:\wp>java override1
Employee Id : 101
Employee Name : senthil
Employee Designation : Professor
Employee Salary : 50000.0
Sports Employee Grade : Batsman
D:\wp>
```

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

**Ex. No : 4. write a Java program to illustrate Constructor Overloading for adding numbers of different data types.**

**Date :**

**Aim**

Write a java program to demonstrate the Constructor Overloading concept.

**Algorithm**

1. Start the program.
2. Create a class consov.
3. Create a method name as class name consov() and pass values into the different argument.
4. Calculate the various data type with values.
5. Display result by using object.
6. Stop the program.

## **Program**

```
class consov
{
    consov(int a, int b)
    {
        System.out.println("\ninteger value : "+(a+b));
    }

    consov(double a, double b)
    {
        System.out.println("\ndouble value : "+(a+b));
    }

    consov(float a, float b)
    {
        System.out.println("\nfloat value : "+(a+b));
    }

    consov(byte a, byte b)
    {
        System.out.println("\nbyte value : "+(a+b));
    }

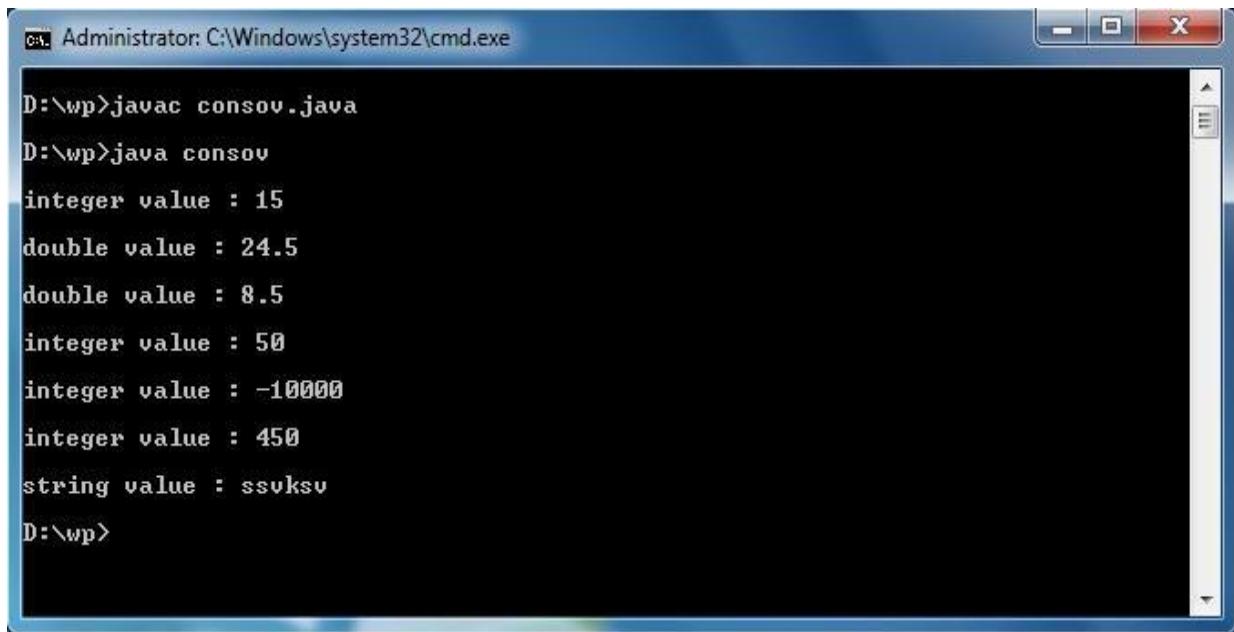
    consov(short a, short b)
    {
        System.out.println("\nshort value : "+(a+b));
    }

    consov(long a, long b)
    {
        System.out.println("\nlong value : "+(a+b));
    }
}
```

```
consov(String a, String b)
{
    System.out.println("\nstring value : "+(a+b));
}

public static void main(String arg[])
{
    consov t1,t2,t3,t4,t5,t6,t7;
    t1 = new consov(5,10);
    t2 = new consov(10.3,14.2);
    t3 = new consov(5.3,3.2);
    t4 = new consov(100,-50);
    t5 = new consov(10000,-20000);
    t6 = new consov(250,200);
    t7 = new consov("ssv","ksv");
}
}
```

**Output :**



D:\wp>javac consov.java  
D:\wp>java consov  
integer value : 15  
double value : 24.5  
double value : 8.5  
integer value : 50  
integer value : -10000  
integer value : 450  
string value : ssuksu  
D:\wp>

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

**Ex. No : 5.** Write a JAVA program using an Interface which has a class called Account that creates the following: Data members: Name of the depositor, Account No, Account Type, Balance amount in the Account. Methods: To assign initial values, to deposit an amount, to withdrawn amount after checking balance and to display the name and balance.

**Date :**

**Aim**

Write a java program to demonstrate Interface concept.

**Algorithm**

1. Start the program.
2. Create an interface account.
3. Create a class emp.
4. Create a method saving, deposit and withdraw.
5. Display the result.
6. Stop the program.

## **Program**

```
interface account
{
    double saving(double ibala);
    double deposit(double amount);
    double withdraw(double amount);
}

class emp implements account
{
    double bala1=0;
    double depo1=0;
    double withd1=0;

    public double saving(double ibala)
    {
        bala1 = ibala;
        return(bala1);
    }

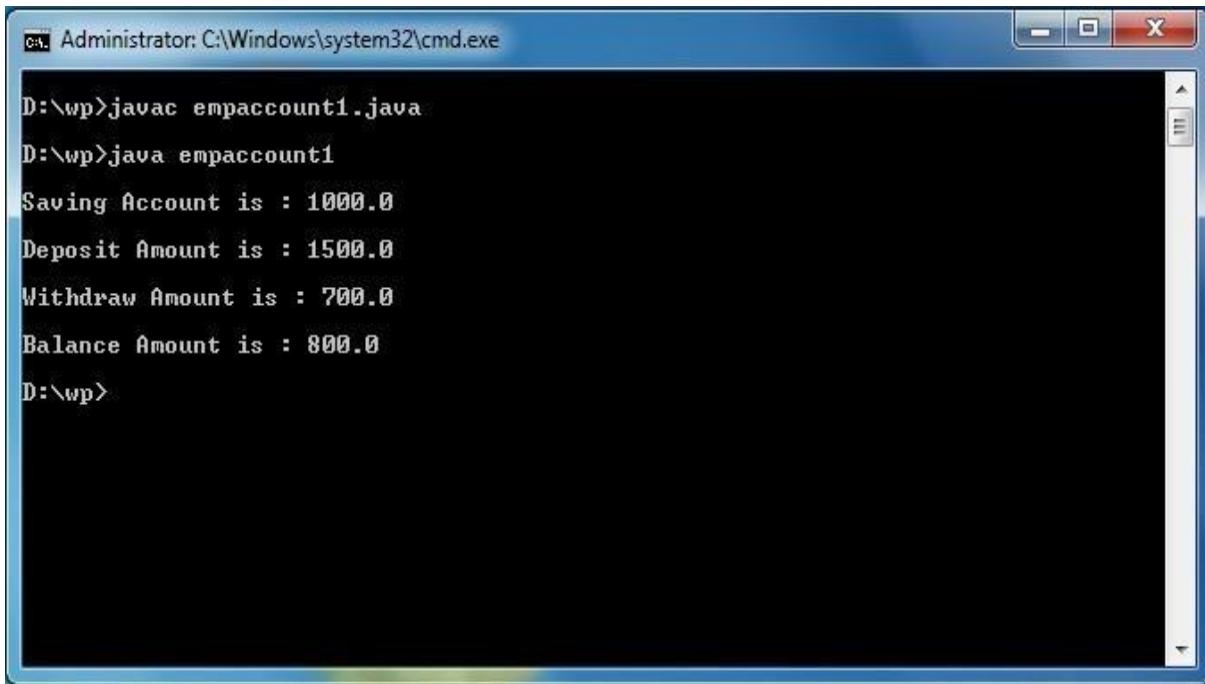
    public double deposit(double amount)
    {
        depo1 = bala1 + amount;
        return(depo1);
    }

    public double withdraw(double amount)
    {
        withd1 = bala1 - amount;
        return(withd1);
    }
}
```

```
public double balance()
{
    return(depo1 - withd1);
}

class empaccount1
{
    public static void main(String arg[])
    {
        emp e = new emp();
        account aa;
        aa = e;
        System.out.println("Saving Account is : "+e.saving(1000));
        System.out.println("Deposit Amount is : " +e.deposit(500));
        System.out.println("Withdraw Amount is : " +e.withdraw(300));
        System.out.println("Balance Amount is : "+e.balance());
    }
}
```

**Output :**



D:\wp>javac empaccount1.java  
D:\wp>java empaccount1  
Saving Account is : 1000.0  
Deposit Amount is : 1500.0  
Withdraw Amount is : 700.0  
Balance Amount is : 800.0  
D:\wp>

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

**Ex. No. : 6.** Develop a Java Program to demonstrate Multilevel Inheritance using the classes Person (Data members – name, age), Teacher (Data members – qualification, salary) and GuestTeacher (Data member – pay per lecture (ppl)). Include method in each class and display the details.

**Date :**

**Aim**

Write a java program to demonstrate Multilevel Inheritance concept.

**Algorithm**

1. Start the program.
2. Create a class person, derived class Teacher and guestTeacher.
3. Initialize the name, age in person class, qualification, salary in teacher class.
4. Calculate the lecture per day in guestTeacher class.
5. Display the result.
6. Stop the program.

## **Program**

```
class Person
{
    String name;
    int age;
    void getdata(String a,int b)
    {
        name=a; age=b;
    }
}

class Teacher extends Person
{
    String qualification;
    float salary;
    void getqual(String s)
    {
        qualification =s;
    }
    void getsal(float a)
    {
        salary=a;
    }
    Void display()
    {
        System.out.println("\nName of Faculty = "+name);
        System.out.println("\nAge of Faculty = "+age);
        System.out.println("\nQualification of Faculty = "+qualification);
        System.out.println("\nSalary of Faculty = "+salary);
    }
}
```

```

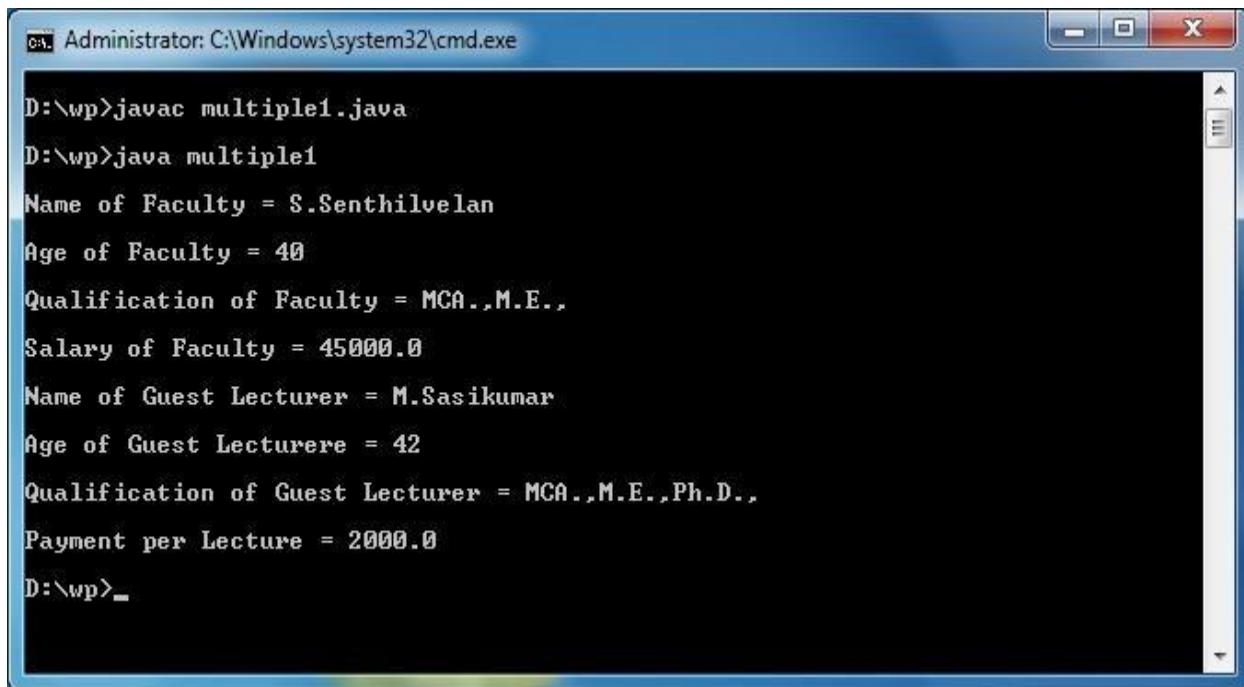
class guestTeacher extends Teacher
{
    float ppl;
    void getppl(float a)
    { ppl=a; }
    void display( )
    {
        System.out.println("\nName of Guest Lecturer = "+name);
        System.out.println("\nAge of Guest Lecturer = "+age);
        System.out.println("\nQualification of Guest Lecturer = "+qualification);
        System.out.println("\nPayment per Lecture = "+ppl);
    }
}

class multiple1
{
    public static void main(String args[])
    {
        Teacher teach=new Teacher();
        teach.getdata("S.Senthilvelan",40);
        teach.getqual("MCA.,M.E.,");
        teach.getsal(45000);
        teach.display( );

        guestTeacher gteach=new guestTeacher();
        gteach.getdata("M.Sasikumar",42);
        gteach.getqual("MCA.,M.E.,Ph.D.,");
        gteach.getppl(2000);
        gteach.display();
    }
}

```

**Output :**



D:\wp>javac multiple1.java  
D:\wp>java multiple1  
Name of Faculty = S.Senthilvelan  
Age of Faculty = 40  
Qualification of Faculty = MCA.,M.E.,  
Salary of Faculty = 45000.0  
Name of Guest Lecturer = M.Sasikumar  
Age of Guest Lecturer = 42  
Qualification of Guest Lecturer = MCA.,M.E.,Ph.D.,  
Payment per Lecture = 2000.0  
D:\wp>\_

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

**Ex. No : 7.** Write a Java program to illustrate Polymorphism for the following: Create a main class Exam. Get the regno, name, mark1, mark2, mark3, from the user. Create a base class Student and display the student details using show() method. Create a derived class placedstudent and display the placement details using show() method.

**Date :**

**Aim**

Write a java program to demonstrate Polymorphism concept.

**Algorithm**

1. Start the program.
2. Create the class PolyExam.
3. Create a base class student inherit all methods, as well as create a derived class.  
placedstudent inherit all method from baseclass.
4. Display the result by using an object.
5. Stop the program.

## **Program**

```
import java.io.*;
import java.util.*;
class student
{
    String nam;
    int reg,ma1,ma2,ma3;

    void show(int re,String na,int m1,int m2,int m3)
    {
        reg = re;  nam = na;  ma1 = m1;  ma2 = m2;  ma3 = m3;
        System.out.println("\nStudents Details");
        System.out.println("\nRegister Number : "+reg);
        System.out.println("\nName : "+nam);
        System.out.println("\nMark1 : "+ma1);
        System.out.println("\nMark2 : "+ma2);
        System.out.println("\nMark3 : "+ma3);
    }
}

class placedstudent extends student
{
    void show(int re,String na,int m1,int m2,int m3)
    {
        reg = re; nam = na; ma1 = m1; ma2 = m2; ma3 = m3;
        System.out.println("\nPlaced Students Details");
        System.out.println("\nRegister Number : "+reg);
        System.out.println("\nName : "+nam);
        System.out.println("\nMark1 : "+ma1);
        System.out.println("\nMark2 : "+ma2);
        System.out.println("\nMark3 : "+ma3);
    }
}
```

```
class PolyExam
{
    public static void main (String arg[]) throws IOException
    {
        String s1,na,s2,s3,s4;
        int re,m1,m2,m3;
        s1 = new String(); s2 = new String(); s3 = new String();
        DataInputStream sc = new DataInputStream(System.in);

        System.out.print("\nEnter the Register Number : ");
        s1 = sc.readLine();
        re = Integer.parseInt(s1);

        System.out.print("\nEnter the Name :");
        na = sc.readLine();

        System.out.print("\nEnter the Mark1 :");
        s2 = sc.readLine();
        m1 = Integer.parseInt(s2);

        System.out.print("\nEnter the Mark2 : ");
        s3 = sc.readLine();
        m2 = Integer.parseInt(s3);

        System.out.print("\nEnter the Mark3 : ");
        s3 = sc.readLine();
        m3 = Integer.parseInt(s3);

        student a1 = new student();
        placedstudent a2 = new placedstudent();

        a1.show(re,na,m1,m2,m3);
        a2.show(re,na,m1,m2,m3);
    }
}
```

## Output :

```
D:\wp>javac PolyExam.java
Note: PolyExam.java uses or overrides a deprecated API.
Note: Recompile with -deprecation for details.

D:\wp>java PolyExam

Enter the Register Number : 101
Enter the Name : vicky
Enter the Mark1 : 75
Enter the Mark2 : 85
Enter the Mark3 : 92
Students Details
Register Number : 101
Name : vicky
Mark1 : 75
Mark2 : 85
Mark3 : 92
Placed Students Details
Register Number : 101
Name : vicky
Mark1 : 75
Mark2 : 85
Mark3 : 92
D:\wp>
```

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

**Ex. No : 8.** Develop a Java program to perform Arithmetic calculations using packages. (a) Create package arithmetic. (b) Create a class calc containing the following methods : add, sub, mul and div (c) Create a main class arith. Get the inputs from the User and print the calculated values by invoking the above methods

**Date :**

**Aim**

Write a java program to demonstrate Package concept.

**Algorithm**

1. Start the program.
2. Create package arithmetic; initiate all information from this package.
3. Create a main class arith and import all the information from package.
4. Display the result.
5. Stop the program.

## **Program**

```
package arithmetic;

public class calc
{
    int a,b;

    public calc(int x,int y)
    {
        a = x;
        b = y;
    }

    public void add()
    {   System.out.println("Addition : "+(a+b));   }

    public void sub()
    {   System.out.println("Subtraction : "+(a-b));   }

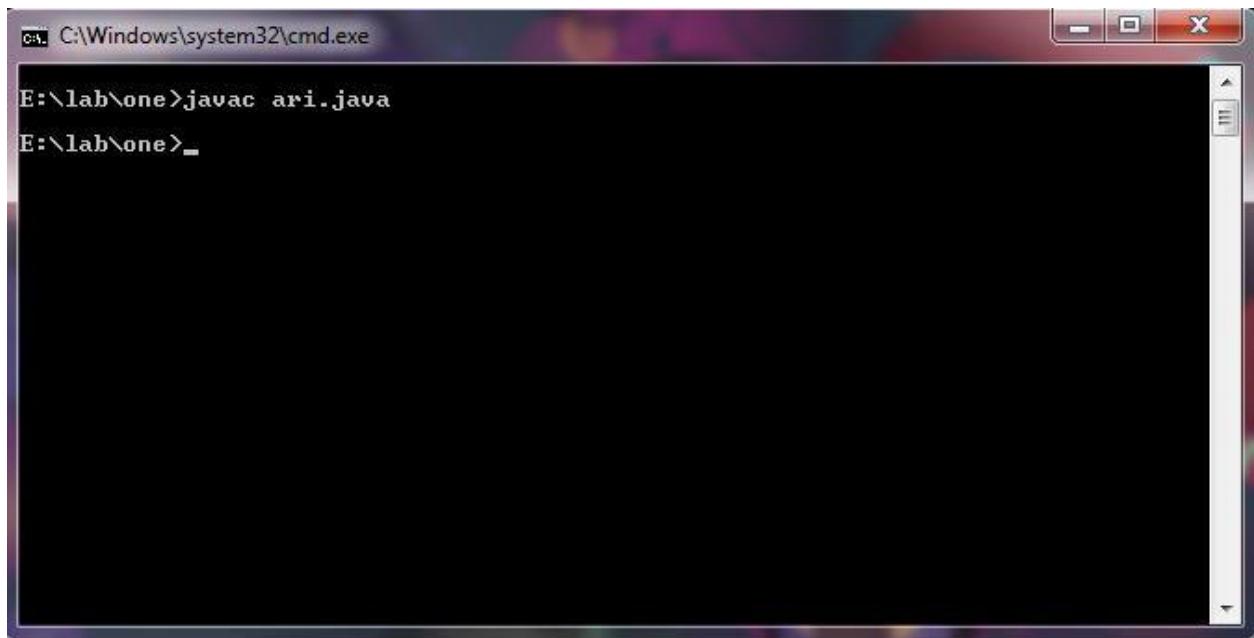
    public void mul()
    {   System.out.println("Multiplication : "+(a*b));   }

    public void div()
    {   System.out.println("Division : "+(a/b));   }
}
```

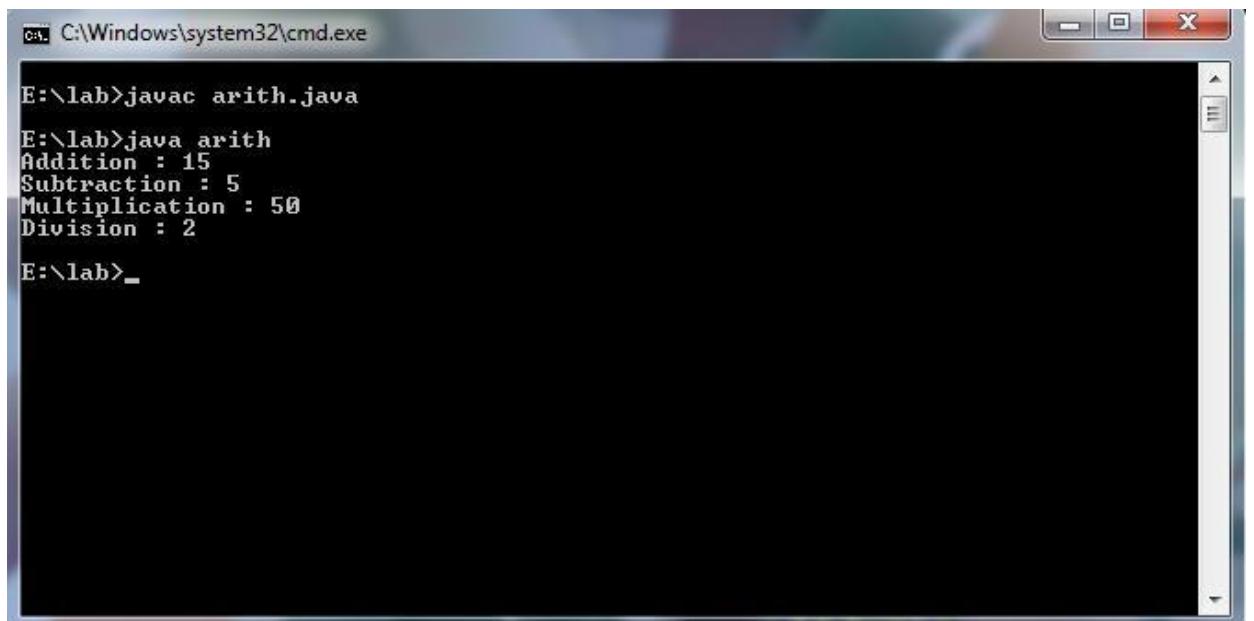
### **Import the package to the main program**

```
import arithmetic.*;  
  
class arith  
{  
    public static void main(String arg[])  
    {  
        calc t =new calc(10,5);  
        t.add();  
        t.sub();  
        t.mul();  
        t.div();  
    }  
}
```

**Output :**



```
C:\Windows\system32\cmd.exe
E:\lab\one>javac ari.java
E:\lab\one>_
```



```
C:\Windows\system32\cmd.exe
E:\lab>javac arith.java
E:\lab>java arith
Addition : 15
Subtraction : 5
Multiplication : 50
Division : 2
E:\lab>_
```

**Ex. No : 9.** Write a Java program to implement the following in-built exceptions:

- (a) `ArrayIndexOutOfBoundsException` (b) `ArithemticException`

**Date:**

**Aim**

Write a java program to demonstrate any two exception handling concept.

**Algorithm**

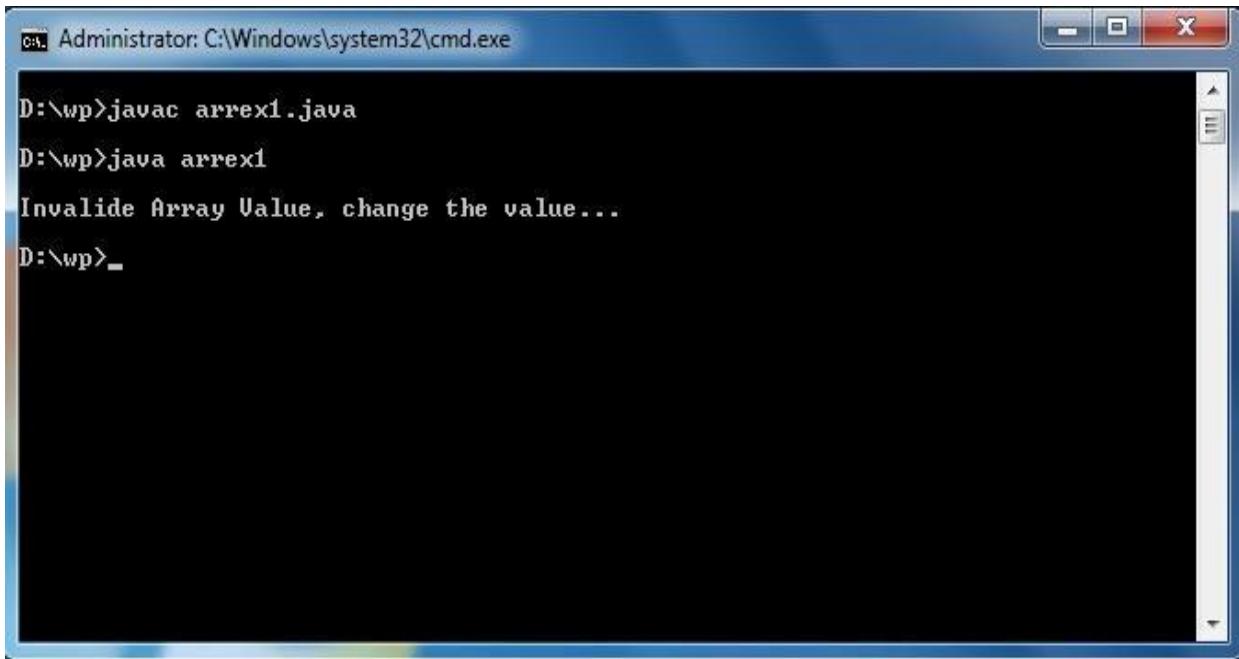
1. Start the program.
2. Create a class and apply the try block for main program.
3. Initialize the values.
4. Calculate invalid array and division by zero value.
5. Apply `ArrayIndexOutOfBoundsException` and `ArithemticException`.
6. Finally print the statement, and end the program.
7. Stop the program.

## **Program**

### **(a) ArrayIndexOutOfBoundsException**

```
class arrex1
{
    public static void main(String arg[])
    {
        try
        {
            int a[] = {2,4,6};
            System.out.println(a[5]);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
        }
        finally
        {
            System.out.println("\nInvalide Array Value, change the value...");
        }
    }
}
```

**Output :**



D:\wp>javac arrex1.java  
D:\wp>java arrex1  
Invalide Array Value, change the value...  
D:\wp>\_

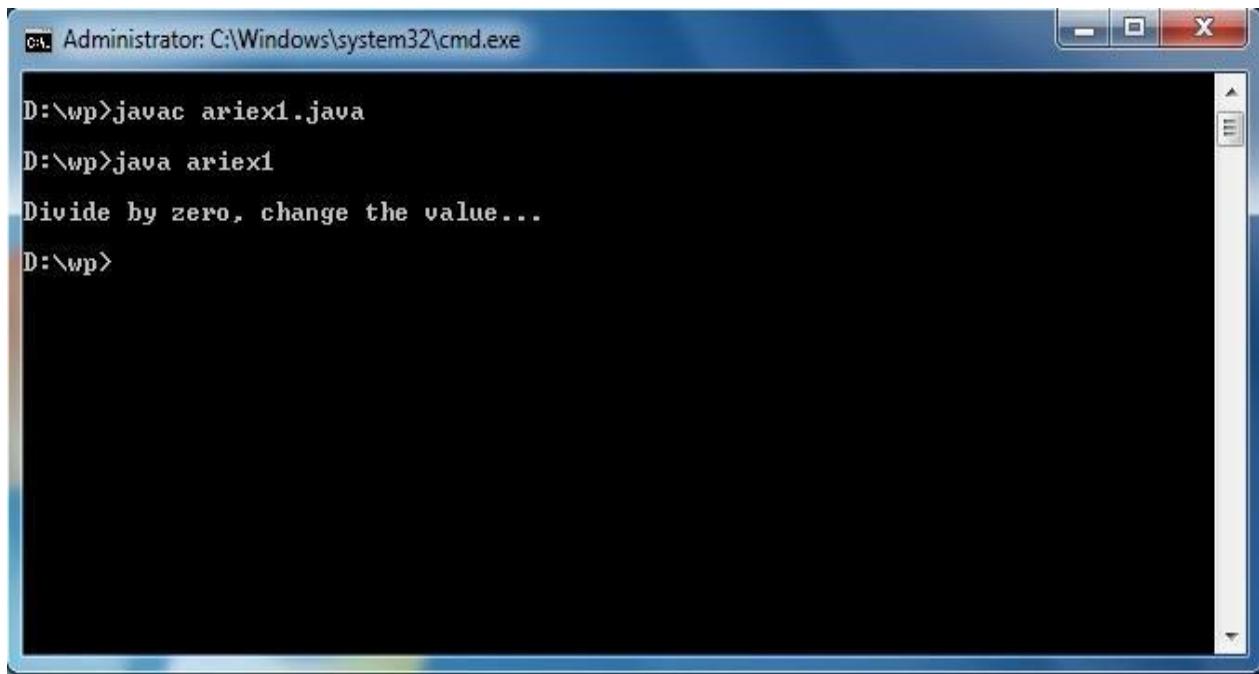
A screenshot of a Windows Command Prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The window has a blue header bar with standard window controls (minimize, maximize, close) on the right. The main body of the window is black and contains white text. The text shows the user entering "javac arrex1.java" followed by "java arrex1". The output message "Invalide Array Value, change the value..." is displayed, indicating a runtime error or warning. The command prompt ends with a final "D:\wp>\_".

## **Program**

### **(b) ArithmeticException**

```
class ariex1
{
    public static void main(String arg[])
    {
        try
        {
            int a,b,c;
            a=10; b=0;
            c=a/b;
            System.out.println(c);
        }
        catch(ArithmeticException e)
        {
        }
        finally
        {
            System.out.println("\nDivide by zero, change the value...");
        }
    }
}
```

**Output :**



A screenshot of a Windows Command Prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The window shows the following text:  
D:\wp>javac ariex1.java  
D:\wp>java ariex1  
Divide by zero, change the value...  
D:\wp>

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

**Ex. No : 10.** Write a Java program that create 3 threads that the 1st thread to display GOOD MORNING for every 1second, 2nd thread to display HELLO for every 2seconds and the 3rd thread to display WELCOME for every 3 seconds.

**Date :**

**Aim**

Write a java program to demonstrate multithreading concept.

**Algorithm**

1. Start the program.
2. Create a class threa3 extends all methods from Thread.
3. Use try and catch into the run() method.
4. Create another class threa2 as well as same process from previous method().
5. Create a main class MyThread.
6. Initiate all the functions from subclass to main class.
7. Display the result.
8. Stop the program.

## **Program**

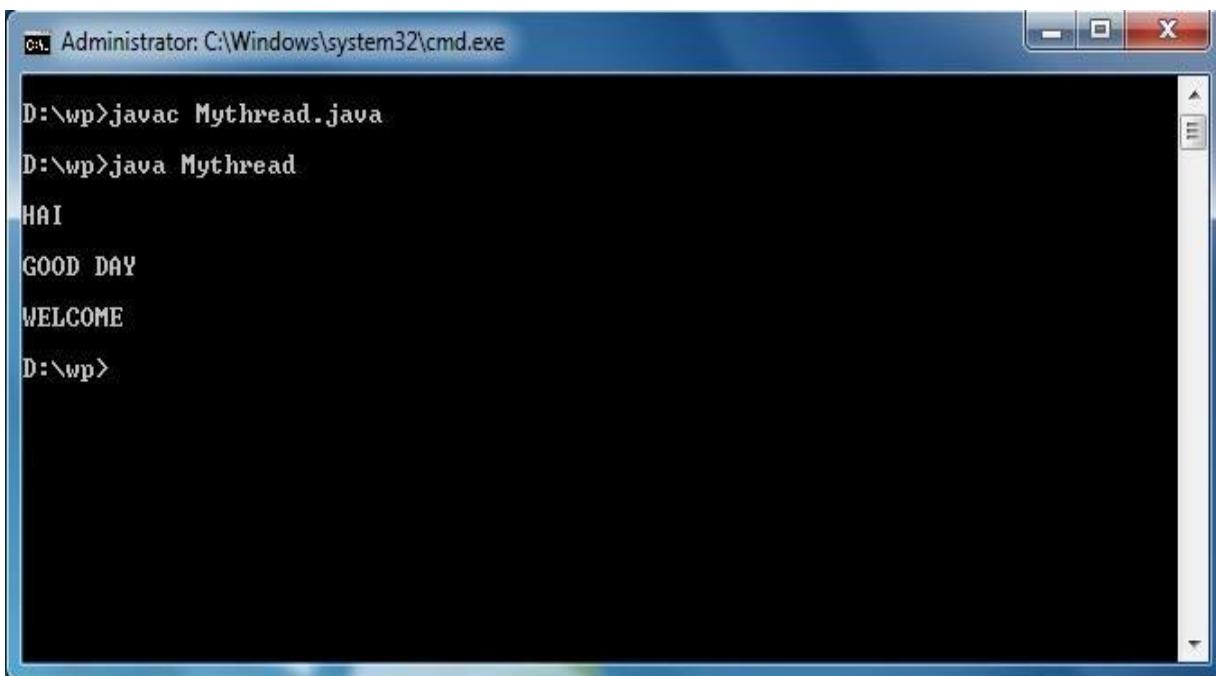
```
class threa3 extends Thread
{
    public void run()
    {
        try
        {
            Thread.sleep(8000);
            System.out.println("\nWELCOME");
        }
        catch(Exceptione)    { }
    }
}
```

```
class threa2 extends Thread
{
    public void run()
    {
        try
        {
            Thread.sleep(4000);
            System.out.println("\nGOOD DAY");
        }
        catch(Exceptione)    { }
    }
}
```

```
class threa1 extends Thread
{
    public void run()
    {
        try
        {
            Thread.sleep(1000);
            System.out.println("\nHAI");
        }
        catch(Exception e) {}
    }
}
```

```
public class Mythread
{
    public static void main(String args[])
    {
        threa3 t3 = new threa3();
        t3.start();
        threa2 t2 = new threa2();
        t2.start();
        threa1 t1 = new threa1();
        t1.start();
    }
}
```

**Output :**



A screenshot of a Windows Command Prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The window shows the following text output:

```
D:\wp>javac Mythread.java
D:\wp>java Mythread
HAI
GOOD DAY
WELCOME
D:\wp>
```

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

**Ex. No : 11. Design an Applet for displaying Polygon when the user presses the Keyboard and display your College name when you release the Keyboard.**

**Date :**

**Aim**

Write a java program to demonstrate key event using Applet.

**Algorithm**

1. Start the program
2. Initialize the array values for polygon.
3. Initialize Graphics and KeyListener in the Applet.
4. Demonstrate getKeyChar() method.
5. Display the Polygon when user pressed any key.
6. Display the content in status bar when user released any key.
7. Stop the program.

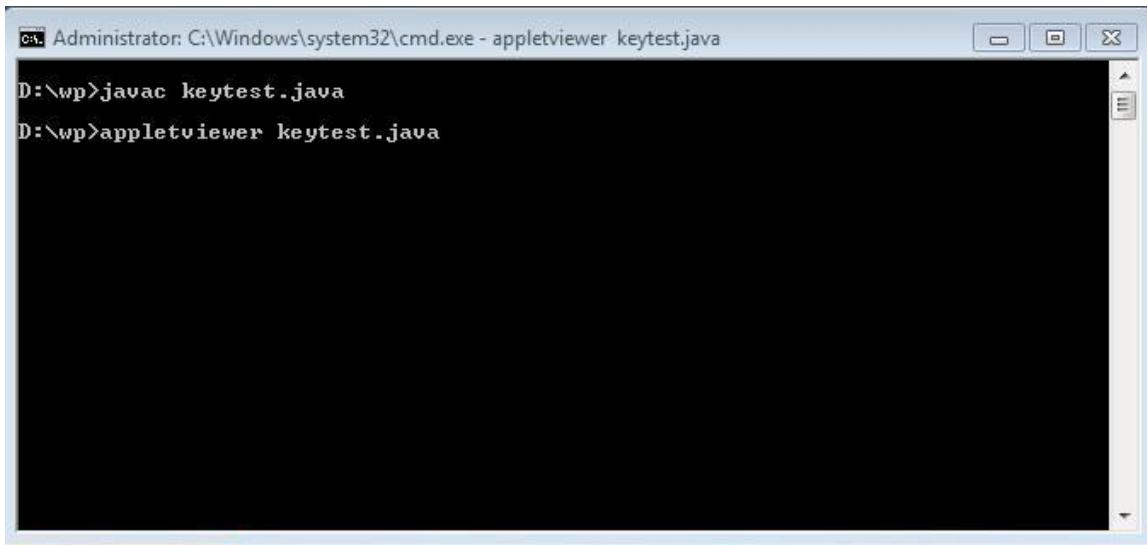
### **Program :**

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

public class keytest extends Applet implements KeyListener
{
    int xa[] = { 120, 125, 150, 150, 200, 200};
    int ya[] = { 175, 100, 100, 175, 175, 200};
    Graphics g;
    public void init()
    {
        g=getGraphics();
        addKeyListener(this);
    }
    public void paint(Graphics g)
    {
        g.drawString("Enter any Character inthe keyboard .... ",20,20);
    }
    public void keyPressed(KeyEvent e)
    {
        g.drawPolygon(xa, ya, 6);
        showStatus("keypressed");
    }
    public void keyReleased(KeyEvent e)
    {
        showStatus("key released - Paavai College of Engineering");
    }
    public void keyTyped(KeyEvent e)
    {
        showStatus("key typed:"+e.getKeyChar());
    }
}

/* <applet code ="keytest.class" width = 350 height =350>
</applet> */
```

## **Output :**



```
Administrator: C:\Windows\system32\cmd.exe - appletviewer keytest.java
D:\wp>javac keytest.java
D:\wp>appletviewer keytest.java
```

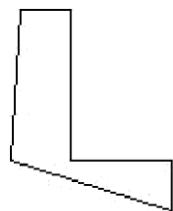


Applet View « keytest.class



Applet

Enter any Character in the keyboard....



key" typed : s

AppletView « keytest.class



Applet

Enter any Character in the keyboard...

PAAVAI ENGINEERING COLLEGE (Autonomous)		
DESCRIPTION	MAX. MARKS	MARKS AWARDED
Preparation & Conduction	<b>20</b>	
Observation & Results	<b>10</b>	
Record Completion	<b>05</b>	
Viva Voce	<b>05</b>	
<b>TOTAL</b>	<b>40</b>	

keyreleased-PaavaiCollegeofEngineering

**Ex. No: 12.** Create a Java Applet that contains three buttons. One button should be labeled Circle, one should be labeled Square and one should be labeled Reset. Display the shapes when the button is pressed.

**Date :**

**Aim**

Write a java program to demonstrate Button Event using Applet.

**Algorithm**

1. Start the program.
2. Initialize Label and Button.
3. Initialize Graphics and ActionListener in the Applet.
4. Demonstrate actionPerformed() method.
5. Display the Circle when user pressed Circle Button.
6. Display the Square when user pressed Square Button.
7. Stop the program

## **Program**

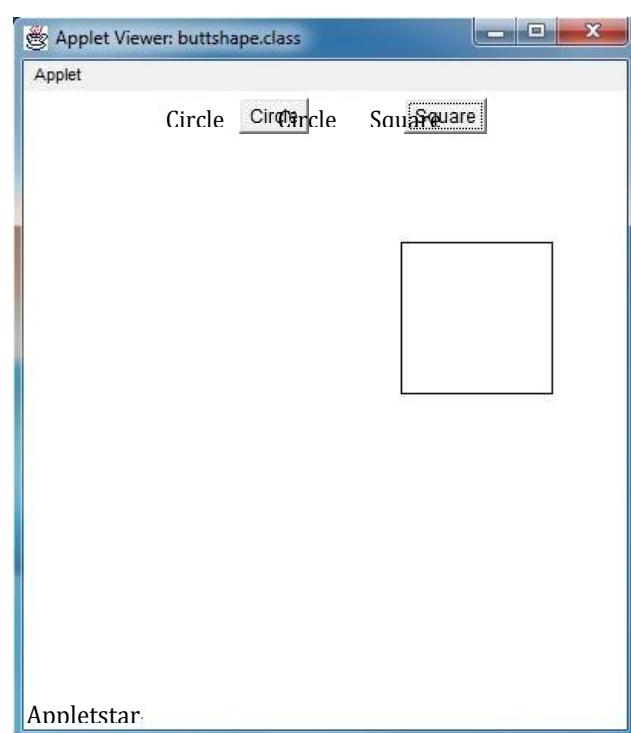
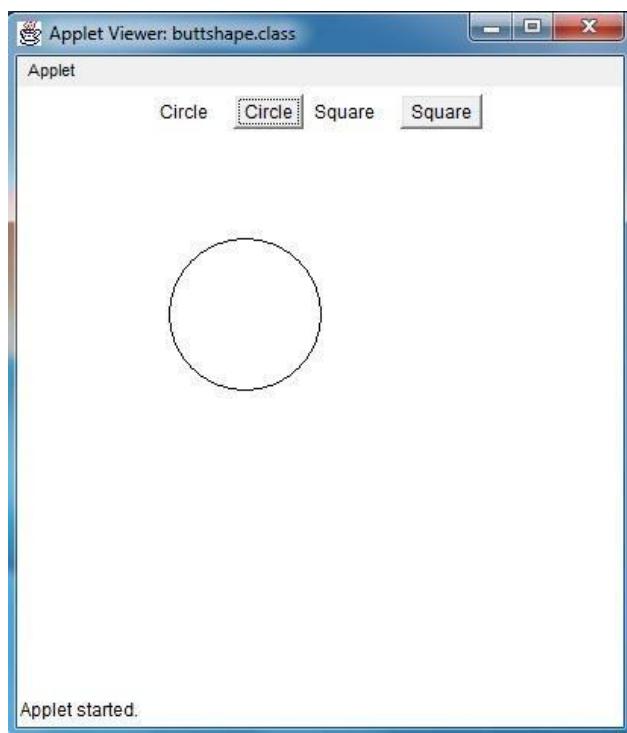
```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class buttshape extends Applet
{
    Label l1,l2;
    Button b1,b2;
    Graphics g;
    public void init()
    {
        l1 =newLabel("Circle");      l2 = new Label("Square");
        b1 =newButton("Circle");     b2 = new Button("Square");
        add(l1); add(b1); add(l2);add(b2);
        b1.addActionListener(new s());   b2.addActionListener(new s());
        g=getGraphics();
    }
    class s implements ActionListener
    {
        public void actionPerformed(ActionEvent e)
        {
            if(e.getSource()==b1)
            {   g.drawOval(100,100,100,100);   }
            if(e.getSource()==b2)
            {   g.drawRect(250,100,100,100);   }
        }
    }
}

/* <applet code = "buttshape.class" height = 400 width = 400>
</applet> */
```

## Output :

```
C:\ Administrator: C:\Windows\system32\cmd.exe - appletviewer buttshape.java
D:\wp>javac buttshape.java
D:\wp>appletviewer buttshape.java
```



**Ex. No : 13. Calculate Factorial Program of given number using RMI concept.**

**Date :**

**Aim**

Write a java Program to calculate factorial of given number using RMI concept.

**Algorithm**

1. Start the Program.
2. Define the interface and server into which contains the declarations at method().
3. Create a method rimp() input in which it will return the if statement arguments.
4. Create a server program in which the class rser implements the interface rint.
5. In the server program create an object for the class rimp interrupt using naming rebind display the object.
6. Create a client program in which object is created for the interface.
7. Using the object call all method supplying the value as run time arguments.
8. Display the result.
9. Stop the program.

## **Program**

### **Interface Program**

#### **rint.java**

```
import java.rmi.*;  
  
public interface rint extends Remote  
{  
    double fact(double x) throws RemoteException;  
}
```

### **Implementation Program**

#### **rimp.java**

```
import java.rmi.*;  
import java.rmi.server.*;  
public class rimp extends UnicastRemoteObject implements rint  
{  
    public rimp() throws RemoteException  
    {}  
    public double fact(double x) throws RemoteException  
    {  
        if(x <= 1)  
            return(1);  
        else  
            return(x*fact(x-1));  
    }  
}
```

## **Server Program**

**rser.java**

```
import java.rmi.*;
import java.net.*;

public class rser
{
    public static void main(String arg[])
    {
        try
        {
            rimp ri = new rimp();

            Naming.rebind("rser",ri);

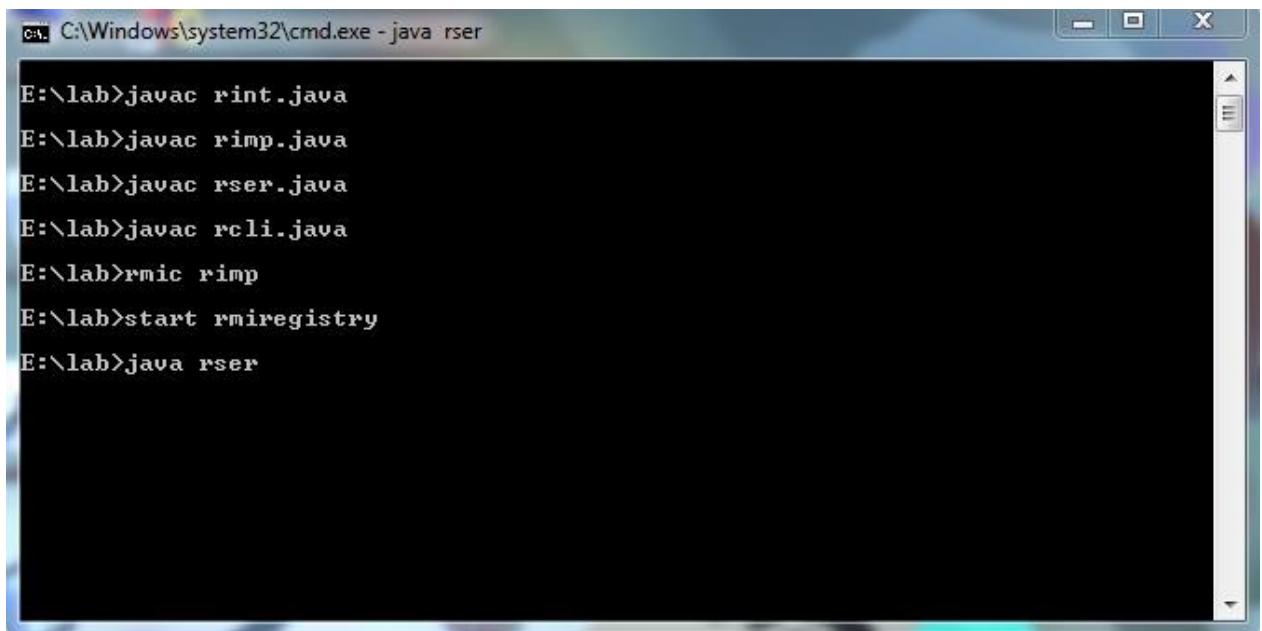
        }catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

## **Client Program**

### **rcli.java**

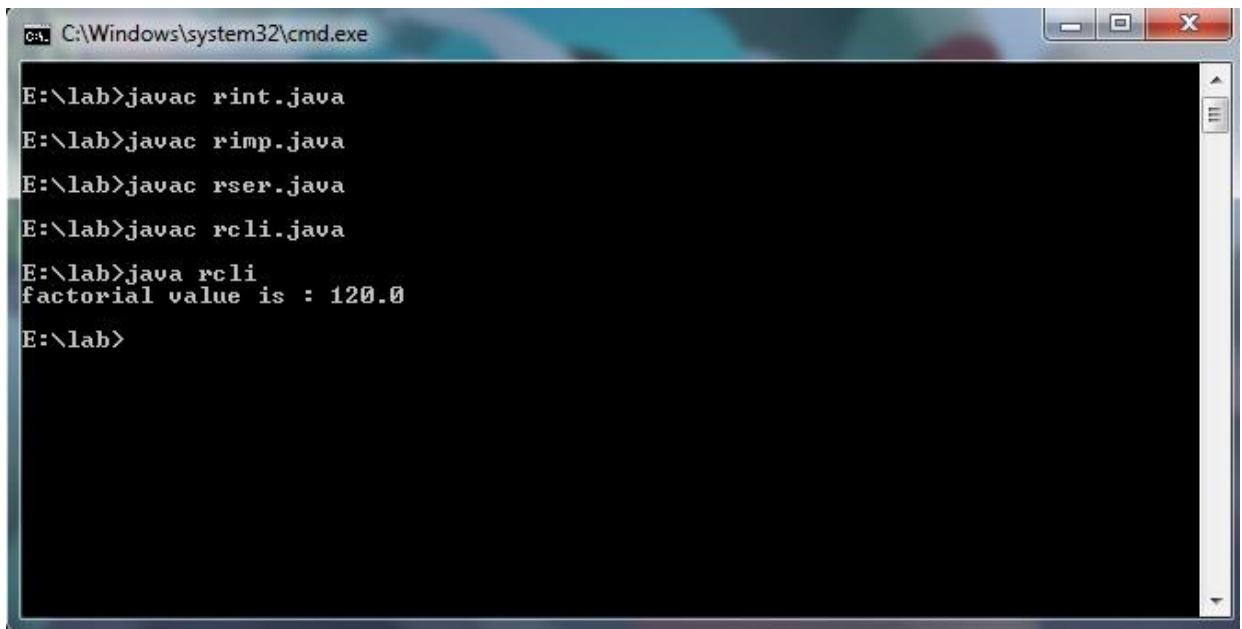
```
import java.rmi.*;  
  
public class rcli  
{  
    public static void main(String arg[])  
    {  
        try  
        {  
            Rint rr = (Rint)Naming.lookup("rmi://172.16.13.2/rser");  
  
            double s = rr.fact(5);  
  
            System.out.println("factorial value is : "+s);  
  
        }catch(Exception e)  
        {  
            System.out.println(e);  
        }  
    }  
}
```

## **Output :**



```
ca C:\Windows\system32\cmd.exe - java rser

E:\lab>javac rint.java
E:\lab>javac rimp.java
E:\lab>javac rser.java
E:\lab>javac rcli.java
E:\lab>rmic rimp
E:\lab>start rmiregistry
E:\lab>java rser
```



```
ca C:\Windows\system32\cmd.exe

E:\lab>javac rint.java
E:\lab>javac rimp.java
E:\lab>javac rser.java
E:\lab>javac rcli.java
E:\lab>java rcli
factorial value is : 120.0
E:\lab>
```

**Ex. No : 14. Develop a Java Program to demonstrate JDBC Connectivity.**

**Date:**

**Aim**

Write a java program demonstrate JDBC concept.

**Algorithm**

1. Start the program
2. Create a class jdat and write the program.
3. Create MS-Access kkk for data base purpose.
4. Create a table kk for storage purpose.
5. Declare Connection and Statement.
6. Create Statement, Execute Statement.
7. Display the output of program
8. Stop the program

## **Program**

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

public class jdat extends Frame
{
    Label l1,l2;
    TextField t1,t2;
    Button b1;

    String url="jdbc:odbc:kkk";
    String query;
    Connection con;
    Statement stmt;

    jdat()
    {
        setLayout(new FlowLayout());
        l1 = new Label("Number : ");
        l2 = new Label("Name : ");
        t1 = new TextField(10);
        t2 = new TextField(10);
        b1 = new Button("add");
        add(l1); add(t1); add(l2); add(t2); add(b1);

        b1.addActionListener(new s());
        addWindowListener(new w());
    }
}
```

```

class w extends WindowAdapter
{
    public void windowClosing(WindowEvent we)
    {   System.exit(0);   }

}

public void cone()
{
    try
    {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    }
    catch(java.lang.ClassNotFoundException ex)
    {
        System.err.println(ex.getMessage());
    }
}
}

class s implements ActionListener
{
    public void actionPerformed(ActionEvent ae)
    {
        query = "select num,nam from ks";
        if(ae.getSource() == b1)
        {
            try
            {
                con = DriverManager.getConnection(url,"","");
                String a,b;
                int c;
                a = t1.getText();

```

```
b = t2.getText();
c = Integer.parseInt(a);
stmt = con.createStatement();

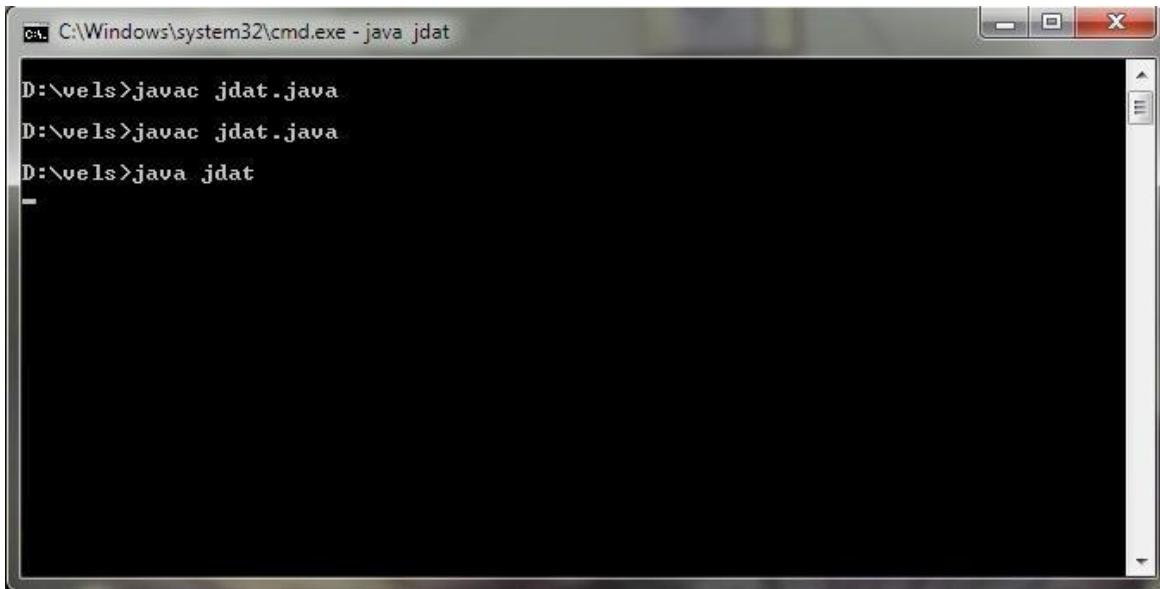
stmt.executeUpdate("insert into kk values("+a+","+b+ ")");
stmt.close();
con.close();
}

catch(SQLException ee)
{
    System.err.println(ee.getMessage());
}
t1.setText("");
t2.setText("");
}
}

}

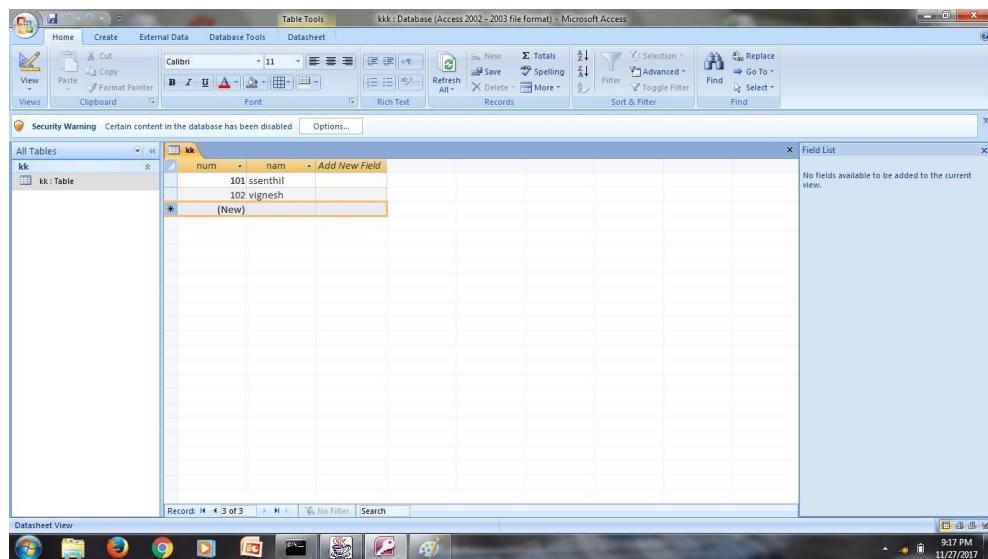
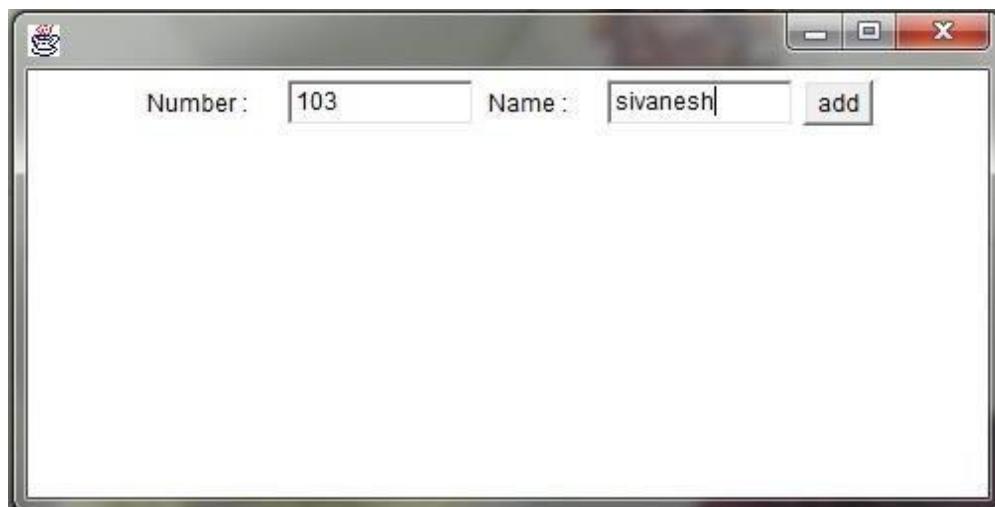
public static void main(String arg[])
{
    jdat k = new jdat();
    k.cone();
    k.show();
    k.setBounds(0,0,500,400);
}
}
```

## Output :



```
C:\Windows\system32\cmd.exe - java jdat

D:\vels>javac jdat.java
D:\vels>javac jdat.java
D:\vels>java jdat
-
```



A screenshot of Microsoft Access Datasheet View showing the "kk" table. The table has two records:

num	nam
101	ssenthil
102	vignesh