

EX. NO : 1
DATE:

ELECTRICITY BILL GENERATION

AIM:

To develop a Java application to generate Electricity bill. Create a class with the following members: Consumer no., consumer name, previous month reading, current month reading, type of EB connection (i.e domestic or commercial). Compute the bill amount using the following tariff.

If the type of the EB connection is domestic, calculate the amount to be paid as follows:

- (⌚) First 100 units - Rs. 1 per unit
- (⌚) 101-200 units - Rs. 2.50 per unit
- (⌚) 201 -500 units - Rs. 4 per unit
- (⌚) > 501 units - Rs. 6 per unit

If the type of the EB connection is commercial, calculate the amount to be paid as follows:

- (⌚) First 100 units - Rs. 2 per unit
- (⌚) 101-200 units - Rs. 4.50 per unit
- (⌚) 201 -500 units - Rs. 6 per unit
- (⌚) > 501 units - Rs. 7 per unit

ALGORITHM:

1. Start
2. Create a class **Ebill** with three member functions: **getdata()**, **calc()**, **display()** and object **ob** is created for another class **Customerdata**.
3. Create another class Customerdata which defines the above functions.
4. Using **getdata()** function get the consumed units for current and previous month .
5. Using **calc()**function calculate the amount depending on the type of EB connection and no. of units consumed.
6. Using **display()** function display the bill.
7. Stop.

PROGRAM:

```
import java.util.*;
class Ebill
{
    public static void main (String args[])
    {
        Customerdata ob = new Customerdata();
        ob.getdata();
        ob.calc();
        ob.display();
    }
}

class Customerdata
{
    Scanner in = new Scanner(System.in);
    Scanner ins = new Scanner(System.in);
    String cname,type;
    int bn;
    double current,previous,tbill,units;
    void getdata()
    {
        System.out.print ("\n\t Enter consumer number ");
        bn = in.nextInt();
        System.out.print ("\n\t Enter Type of connection (D for Domestic or C for Commercial )");
        type = ins.nextLine();
        System.out.print ("\n\t Enter consumer name ");
        cname = ins.nextLine();
        System.out.print ("\n\t Enter previous month reading ");
        previous= in.nextDouble();
        System.out.print ("\n\t Enter current month reading ");
        current= in.nextDouble();
    }
}
```

```

void calc()
{
    units=current-previous;
    if(type.equals("D"))
    {
        if (units<=100)
            tbill=1 * units;
        else if (units>100 && units<=200)
            tbill=2.50*units;
        else if(units>200 && units<=500)
            tbill= 4*units;
        else
            tbill= 6*units;
    }
    else
    {
        if (units<=100)
            tbill= 2 * units;
        else if(units>100 && units<=200)
            tbill=4.50*units;
        else if(units>200 && units<=500)
            tbill= 6*units;
        else
            tbill= 7*units;
    }
}
void display()
{
    System.out.println("\n\t Consumer number = "+bn);
    System.out.println ("\n\t Consumer name = "+cname);
    if(type.equals("D"))
        System.out.println ("\n\t type of connection = DOMESTIC ");
    else
        System.out.println ("\n\t type of connection = COMMERCIAL ");
    System.out.println ("\n\t Current Month Reading = "+current);
    System.out.println ("\n\t Previous Month Reading = "+previous);
    System.out.println ("\n\t Total units = "+units);
    System.out.println ("\n\t Total bill = RS "+tbill);
}
}

```

OUTPUT:

```

D:\>javac Ebill.java
D:\>java Ebill
Enter consumer number 1001
Enter Type of connection (D for Domestic or C for Commercial) D
Enter consumer name Sachin
Enter previous month reading 3000
Enter current month reading 4000
Consumer number = 1001
Consumer name = Sachin
type of connection = DOMESTIC
Current Month Reading = 4000.0
Previous Month Reading = 3000.0
Total units = 1000.0
Total bill = RS 6000.0

```

RESULT:

Thus the java program to generate electricity bill was implemented, executed successfully and the output was verified.

EX NO : 2 CURRENCY CONVERTER, DISTANCE CONVERTER AND TIME CONVERTER
IMPLEMENTATION USING PACKAGES

DATE:

AIM:

To develop a java application to implement currency converter (Dollar to INR, EURO to INR, Yen to INR and vice versa), distance converter (meter to KM, miles to KM and vice versa) , time converter (hours to minutes, seconds and vice versa) using packages.

ALGORITHM:

1. Start
2. Create a Package **currencyconversion** and place the class **currency** under the package
3. Create the methods to perform currency conversion from dollar to rupee, rupee to dollar, euro to rupee, rupee to euro, yen to rupee and rupee to yen.
4. Create the package **distanceconverion** and create the class **distance** within the package
5. Create the methods to convert from meter to km, km to meter, miles to km, km to miles
6. Create the package **timeconversion** and create the class **timer** .Create the methods to convert from hours to minutes ,hours to seconds , minutes to hours and seconds to hours
7. Create a class **converter** and import the packages currencyconversion, distanceconversion and time conversion. Create the objects for the class currency, distance and timer.
8. Get the choice from the user and invoke the methods to perform the corresponding conversion and display the value.
9. Stop

PROGRAM:

currency.java

```
package currencyconversion;
import java.util.*;
public class currency
{
    double inr,usd;
    double euro,yen;
    Scanner in=new Scanner(System.in);
    public void dollartorupee()
    {
        System.out.println("Enter dollars to convert into Rupees:");
        usd=in.nextInt();
        inr=usd*67;
        System.out.println("Dollar =" +usd+"equal to INR=" +inr);
    }
    public void rupeetodollar()
    {
        System.out.println("Enter Rupee to convert into Dollars:");
        inr=in.nextInt();
        usd=inr/67;
        System.out.println("Rupee =" +inr+"equal to Dollars=" +usd);
    }
    public void euotorupee()
    {
        System.out.println("Enter euro to convert into Rupees:");
        euro=in.nextInt();
        inr=euro*79.50;
        System.out.println("Euro =" +euro +"equal to INR=" +inr);
    }
    public void rupeetoeuro()
    {
        System.out.println("Enter Rupees to convert into Euro:");
        inr=in.nextInt();
        euro=(inr/79.50);
        System.out.println("Rupee =" +inr +"equal to Euro=" +euro);
    }
}
```

```

public void yentorupee()
{
    System.out.println("Enter yen to convert into Rupees:");
    yen=in.nextInt();
    inr=yen*0.61;
    System.out.println("YEN="+yen +"equal to INR="+inr);
}
public void rupeetoyen()
{
    System.out.println("Enter Rupees to convert into Yen:");
    inr=in.nextInt();
    yen=(inr/0.61);
    System.out.println("INR="+inr +"equal to YEN"+yen);
}
}

```

distance.java

```

package distanceconversion;
import java.util.*;
public class distance
{
    double km,m,miles;
    Scanner sc = new Scanner(System.in);
    public void kmtom()
    {
        System.out.print("Enter in km ");
        km=sc.nextDouble();
        m=(km*1000);
        System.out.println(km+"km" +"equal to"+m+"metres");
    }
    public void mtokm()
    {
        System.out.print("Enter in meter ");
        m=sc.nextDouble();
        km=(m/1000);
        System.out.println(m+"m" +"equal to"+km+"kilometres");
    }
    public void milestokm()
    {
        System.out.print("Enter in miles");
        miles=sc.nextDouble();
        km=(miles*1.60934);
        System.out.println(miles+"miles" +"equal to"+km+"kilometres");
    }
    public void kmtomiles()
    {
        System.out.print("Enter in km");
        km=sc.nextDouble();
        miles=(km*0.621371);
        System.out.println(km+"km" +"equal to"+miles+"miles");
    }
}

```

timer.java

```

package timeconversion;
import java.util.*;
public class timer
{
    int hours,seconds,minutes;
    int input;
    Scanner sc = new Scanner(System.in);
    public void secondstohours()
    {
        System.out.print("Enter the number of seconds: ");
        input = sc.nextInt();
    }
}

```

```

        hours = input / 3600;
        minutes = (input % 3600) / 60;
        seconds = (input % 3600) % 60;
        System.out.println("Hours: " + hours);
        System.out.println("Minutes: " + minutes);
        System.out.println("Seconds: " + seconds);
    }
    public void minutestohours()
    {
        System.out.print("Enter the number of minutes: ");
        minutes=sc.nextInt();
        hours=minutes/60;
        minutes=minutes%60;
        System.out.println("Hours: " + hours);
        System.out.println("Minutes: " + minutes);
    }
    public void hourstominutes()
    {
        System.out.println("enter the no of hours");
        hours=sc.nextInt();
        minutes=(hours*60);
        System.out.println("Minutes: " + minutes);
    }
    public void hourstoseconds()
    {
        System.out.println("enter the no of hours");
        hours=sc.nextInt();
        seconds=(hours*3600);
        System.out.println("Minutes: " + seconds);
    }
}

```

converter.java

```

import java.util.*;
import java.io.*;
import currencyconversion.*;
import distanceconversion.*;
import timeconversion.*;
class converter
{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        int choice,ch;
        currency c=new currency();
        distance d=new distance();
        timer t=new timer();
        do
        {
            System.out.println("1.dollar to rupee ");
            System.out.println("2.rupee to dollar ");
            System.out.println("3.Euro to rupee ");
            System.out.println("4..rupee to Euro ");
            System.out.println("5.Yen to rupee ");
            System.out.println("6.Rupee to Yen ");
            System.out.println("7.Meter to kilometer ");
            System.out.println("8.kilometer to meter ");
            System.out.println("9.Miles to kilometer ");
            System.out.println("10.kilometer to miles");
            System.out.println("11.Hours to Minutes");
            System.out.println("12.Hours to Seconds");
            System.out.println("13.Seconds to Hours");
            System.out.println("14.Minutes to Hours");
            System.out.println("Enter ur choice");
            choice=s.nextInt();

```

```

switch(choice)
{
    case 1:
    {
        c.dollartorupee();
        break;
    }
    case 2:
    {
        c.rupeetodollar();
        break;
    }
    case 3:
    {
        c.eurotorupee();
        break;
    }
    case 4:
    {
        c.ruepeetoeuro();
        break;
    }
    case 5:
    {
        c.yentorupee();
        break;
    }
    case 6:
    {
        c.ruepetoyen();
        break;
    }
    case 7:
    {
        d.mtokm();
        break;
    }
    case 8:
    {
        d.kmtom();
        break;
    }
    case 9:
    {
        d.milestokm();
        break;
    }
    case 10 :
    {
        d.kmtomiles();
        break;
    }
    case 11:
    {
        t.hourstominutes();
        break;
    }
    case 12:
    {
        t.hourstoseconds();
        break;
    }
}

```

```

        case 13:
    {
        t.secondsstohours();
        break;
    }
    case 14:
    {
        t minutestohours();
        break;
    }
}
System.out.println("Enter 0 to quit and 1 to continue ");
ch=s.nextInt();
}
while(ch==1);
}
}

```

OUTPUT:

```

E:\PROGRAMS>javac converter.java
E:\PROGRAMS>java converter
1.dollar to rupee
2.rupee to dollar
3.Euro to rupee
4..rupee to Euro
5.Yen to rupee
6.Rupee to Yen
7.Meter to kilometer
8.kilometer to meter
9.Miles to kilometer
10.kilometer to miles
11.Hours to Minutes
12.Hours to Seconds
13.Seconds to Hours
14.Minutes to Hours
Enter ur choice
1
Enter dollars to convert into Rupees:
500
Dollar =500.0equal to INR=33500.0
Enter 0 to quit and 1 to continue

```

```

500
Dollar =500.0equal to INR=33500.0
Enter 0 to quit and 1 to continue
1
1.dollar to rupee
2.rupee to dollar
3.Euro to rupee
4..rupee to Euro
5.Yen to rupee
6.Rupee to Yen
7.Meter to kilometer
8.kilometer to meter
9.Miles to kilometer
10.kilometer to miles
11.Hours to Minutes
12.Hours to Seconds
13.Seconds to Hours
14.Minutes to Hours
Enter ur choice
8
Enter in km
2
2.0kmegual to2000.0metres
Enter 0 to quit and 1 to continue

```

RESULT

Thus the Java application to implement currency converter, distance converter and time converter was implemented, executed successfully and the output was verified.

EX NO: 3
DATE:

PAYSLIP GENERATION USING INHERITANCE

AIM:

To develop a java application to generate pay slip for different category of employees using the concept of **inheritance**.

ALGORITHM:

1. Start
2. Create the class **Employee** with name, Empid, address, mailid, mobileno as data members.
3. Inherit the classes **Programmer**, **Asstprofessor**, **Associateprofessor** and **Professor** from employee class.
4. Add Basic Pay (BP) as the member of all the inherited classes.
5. Calculate DA as 97% of BP, HRA as 10% of BP, PF as 12% of BP, Staff club fund as 0.1% of BP.
6. Calculate gross salary and net salary.
7. Generate payslip for all categories of employees.
8. Create the objects for the inherited classes and invoke the necessary methods to display the Payslip
9. Stop

PROGRAM:

Salary.java

```
import java.util.*;
class Employee
{
    int empid;
    long mobile;
    String name, address, mailid;
    Scanner get = new Scanner(System.in);
    void getdata()
    {
        System.out.println("Enter Name of the Employee");
        name = get.nextLine();
        System.out.println("Enter Mail id");
        mailid = get.nextLine();
        System.out.println("Enter Address of the Employee:");
        address = get.nextLine();
        System.out.println("Enter employee id ");
        empid = get.nextInt();
        System.out.println("Enter Mobile Number");
        mobile = get.nextLong();
    }
    void display()
    {
        System.out.println("Employee Name: "+name);
        System.out.println("Employee id : "+empid);
        System.out.println("Mail id : "+mailid);
        System.out.println("Address: "+address);
        System.out.println("Mobile Number: "+mobile);
    }
}

class Programmer extends Employee
{
    double salary,bp,da,hra,pf,club,net,gross;
    void getprogrammer()
    {
        System.out.println("Enter basic pay");
        bp = get.nextDouble();
    }
}
```

```

void calculateprog()
{
    da=(0.97*bp);
    hra=(0.10*bp);
    pf=(0.12*bp);
    club=(0.1*bp);
    gross=(bp+da+hra);
    net=(gross-pf-club);
    System.out.println("*****");
    System.out.println("PAY SLIP FOR PROGRAMMER");
    System.out.println("*****");
    System.out.println("Basic Pay: Rs. "+bp);
    System.out.println("DA: Rs. "+da);
    System.out.println("HRA: Rs. "+hra);
    System.out.println("PF: Rs. "+pf);
    System.out.println("CLUB: Rs. "+club);
    System.out.println("GROSS PAY: Rs. "+gross);
    System.out.println("NET PAY: Rs. "+net);
}
}

class Asstprofessor extends Employee
{
    double salary,bp,da,hra,pf,club,net,gross;
    void getasst()
    {
        System.out.println("Enter basic pay");
        bp = get.nextDouble();
    }
    void calculateasst()
    {
        da=(0.97*bp);
        hra=(0.10*bp);
        pf=(0.12*bp);
        club=(0.1*bp);
        gross=(bp+da+hra);
        net=(gross-pf-club);
        System.out.println("*****");
        System.out.println("PAY SLIP FOR ASSISTANT PROFESSOR");
        System.out.println("*****");
        System.out.println("Basic Pay: Rs. "+bp);
        System.out.println("DA: Rs. "+da);
        System.out.println("HRA: Rs. "+hra);
        System.out.println("PF: Rs. "+pf);
        System.out.println("CLUB: Rs. "+club);
        System.out.println("GROSS PAY: Rs. "+gross);
        System.out.println("NET PAY: Rs. "+net);
    }
}

class Associateprofessor extends Employee
{
    double salary,bp,da,hra,pf,club,net,gross;
    void getassociate()
    {
        System.out.println("Enter basic pay");
        bp = get.nextDouble();
    }
    void calculateassociate()
    {
        da=(0.97*bp);
        hra=(0.10*bp);
        pf=(0.12*bp);
        club=(0.1*bp);
        gross=(bp+da+hra);
        net=(gross-pf-club);
    }
}

```

```

        System.out.println("*****");
        System.out.println("PAY SLIP FOR ASSOCIATE PROFESSOR");
        System.out.println("*****");
        System.out.println("Basic Pay: Rs. "+bp);
        System.out.println("DA: Rs. "+da);
        System.out.println("HRA: Rs. "+hra);
        System.out.println("PF: Rs. "+pf);
        System.out.println("CLUB: Rs. "+club);
        System.out.println("GROSS PAY: Rs. "+gross);
        System.out.println("NET PAY: Rs. "+net);
    }
}

class Professor extends Employee
{
    double salary,bp,da,hra,pf,club,net,gross;
    void getprofessor()
    {
        System.out.println("Enter basic pay");
        bp = get.nextDouble();
    }
    void calculateprofessor()
    {
        da=(0.97*bp);
        hra=(0.10*bp);
        pf=(0.12*bp);
        club=(0.1*bp);
        gross=(bp+da+hra);
        net=(gross-pf-club);
        System.out.println("*****");
        System.out.println("PAY SLIP FOR PROFESSOR");
        System.out.println("*****");
        System.out.println("Basic Pay: Rs. "+bp);
        System.out.println("DA: Rs. "+da);
        System.out.println("HRA: Rs. "+hra);
        System.out.println("PF: Rs. "+pf);
        System.out.println("CLUB: Rs. "+club);
        System.out.println("GROSS PAY: Rs. "+gross);
        System.out.println("NET PAY: Rs. "+net);
    }
}

class Salary
{
    public static void main(String args[])
    {
        int choice,cont;
        do
        {
            System.out.println("PAYROLL");
            System.out.println(" 1.PROGRAMMER \t 2.ASSISTANT PROFESSOR \t 3.ASSOCIATE
PROFESSOR \t 4.PROFESSOR ");
            Scanner c = new Scanner(System.in);
            System.out.print("Enter Your Choice:");
            choice=c.nextInt();
            switch(choice)
            {
                case 1:
                {
                    Programmer p=new Programmer();
                    p.getdata();
                    p.getprogrammer();
                    p.display();
                    p.calculateprog();
                    break;
                }
            }
        }
    }
}

```

```

        case 2:
    {
        Asstprofessor asst=new Asstprofessor();
        asst.getdata();
        asst.getasst();
        asst.display();
        asst.calculateasst();
        break;
    }
    case 3:
    {
        Associateprofessor asso=new Associateprofessor();
        asso.getdata();
        asso.getassociate();
        asso.display();
        asso.calculateassociate();
        break;
    }
    case 4:
    {
        Professor prof=new Professor();
        prof.getdata();
        prof.getprofessor();
        prof.display();
        prof.calculateprofessor();
        break;
    }
}
System.out.print("Please enter 0 to quit and 1 to continue: ");
cont=c.nextInt();
}while(cont==1);
}
}

```

OUTPUT

1.PROGRAMMER 2.ASSISTANT PROFESSOR 3.ASSOCIATE PROFESSOR 4.PROFESSOR

Enter Your Choice: 2

Enter Name of the Employee: Arun K

Enter Mail id: arun007@gmail.com

Enter Address of the Employee: 12, Anna Nagar, Chennai-65

Enter employee id: 5002

Enter Mobile Number: 9876543210

Enter basic pay : 20000

Employee Name: Arun K

Employee id : 5002

Mail id : arun007@gmail.com

Address: 12, Anna Nagar, Chennai-65

Mobile Number: 9876543210

PAY SLIP FOR ASSISTANT PROFESSOR

Basic Pay:Rs. 20000.0

DA:Rs. 19400.0

HRA:Rs. 2000.0

PF:Rs. 2400.0

CLUB:Rs. 2000.0

GROSS PAY:Rs. 41400.0

NET PAY:Rs. 37000.0

Please enter 0 to quit and 1 to continue: 0

RESULT

Thus the Java application to generate pay slip for different category of employees was implemented using inheritance and the program was executed successfully.

EX NO: 4
DATE:

STACK ADT IMPLEMENTATION USING INHERITANCE (INTERFACE)

AIM

To design a Java application to implement array implementation of stack using the concept of **Interface** and **Exception handling**.

ALGORITHM

1. Start
2. Create the interface **Stackoperation** with method declarations for push and pop.
3. Create the class **Astack** which implements the interface and provides implementation for the methods push and pop. Also define the method for displaying the values stored in the stack. Handle the stack overflow and stack underflow condition.
4. Create the class **teststack**. Get the choice from the user for the operation to be performed and also handle the exception that occur while performing the stack operation.
5. Create the object and invoke the method for push, pop, display based on the input from the user.
6. Stop.

PROGRAM

Teststack.java

```
import java.io.*;
interface Stackoperation
{
    public void push(int i);
    public void pop();
}

class Astack implements Stackoperation
{
    int stack[] = new int[5];
    int top=-1;
    int i;
    public void push(int item)
    {
        if(top>=4)
        {
            System.out.println("Overflow");
        }
        else
        {
            top=top+1;
            stack[top]=item;
            System.out.print("Element pushed: "+stack[top]);
        }
    }
    public void pop()
    {
        if(top<0)
            System.out.println("Underflow");
        else
        {
            System.out.print("Element popped: "+stack[top]);
            top=top-1;
        }
    }
}
```

```

public void display()
{
    if(top<0)
        System.out.println("No Element in stack");
    else
    {
        for(i=0;i<=top;i++)
            System.out.println("Element:"+stack[i]);
    }
}

class Teststack
{
    public static void main(String args[]) throws IOException
    {
        int ch,c;
        int i;
        Astack s=new Astack();
        DataInputStream in=new DataInputStream(System.in);
        do
        {
            try
            {
                System.out.println("ARRAY STACK");
                System.out.println("1.Push 2.Pop 3.Display 4.Exit");
                System.out.print("Enter your Choice:");
                ch=Integer.parseInt(in.readLine());
                switch(ch)
                {
                    case 1:
                        System.out.print("Enter the value to push:");
                        i=Integer.parseInt(in.readLine());
                        s.push(i);
                        break;
                    case 2:
                        s.pop();
                        break;
                    case 3:
                        System.out.println("The elements are: ");
                        s.display();
                        break;
                    default:
                        break;
                }
            }
            catch(IOException e)
            {
                System.out.println("IO Error");
            }
            System.out.println("Please enter 0 to quit and 1 to continue ");
            c=Integer.parseInt(in.readLine());
        }while(c==1);
    }
}

```

OUTPUT

```
D:\>javac Teststack.java
D:\>java Teststack
ARRAY STACK
1.Push 2.Pop 3.Display 4.Exit
Enter your Choice: 1
Enter the value to push: 10
Element pushed: 10
Please enter 0 to quit and 1 to continue
1
ARRAY STACK
1.Push 2.Pop 3.Display 4.Exit
Enter your Choice: 1
Enter the value to push: 20
Element pushed: 20
Please enter 0 to quit and 1 to continue
1
ARRAY STACK
1.Push 2.Pop 3.Display 4.Exit
Enter your Choice: 1
Enter the value to push: 30
Element pushed: 30
Please enter 0 to quit and 1 to continue
1
ARRAY STACK
1.Push 2.Pop 3.Display 4.Exit
Enter your Choice: 3
The elements are:
Element: 10
Element: 20
Element: 30
Please enter 0 to quit and 1 to continue
1
ARRAY STACK
1.Push 2.Pop 3.Display 4.Exit
Enter your Choice: 2
Element popped: 30
Please enter 0 to quit and 1 to continue
1
ARRAY STACK
1.Push 2.Pop 3.Display 4.Exit
Enter your Choice: 3
The elements are:
element:10
element:20
Please enter 0 to quit and 1 to continue
0
```

RESULT

Thus the Java application for ADT stack operations has been implemented and executed successfully.

EX NO: 5

DATE:

STRING OPERATIONS USING ARRAY LIST

AIM

To write a java program to perform string operations using ArrayList for the following functions:

- a. Append - add at end
- b. Insert – add at particular index
- c. Search
- d. List all string starts with given letter

ALGORITHM:

1. Start
2. Create the class **ArrayListexample**. Create the object for the **ArrayList** class.
3. Display the options to the user for performing string handling .
4. Use the function `add()` to append the string at the end and to insert the string at the particular index.
5. The function `sort ()` is used to sort the elements in the array list.
6. The function `indexof()` is used to search whether the element is in the array list or not.
7. The function `startsWith ()` is used to find whether the element starts with the specified character.
8. The function `remove()` is used to remove the element from the arraylist.
9. The function `size()` is used to determine the number of elements in the array list.
10. Stop.

PROGRAM:

ArrayListexample.java

```
import java.util.*;
import java.io.*;
public class ArrayListexample
{
    public static void main(String args[]) throws IOException
    {
        ArrayList<String> obj = new ArrayList<String>();
        DataInputStream in=new DataInputStream(System.in);
        int c,ch;
        int i,j;
        String str,str1;
        do
        {
            System.out.println("STRING MANIPULATION");
            System.out.println("*****");
            System.out.println("1. Append at end \t 2.Insert at particular index \t 3.Search \t");
            System.out.println("4. List string that starting with letter \t");
            System.out.println("5. Size \t 6.Remove \t 7.Sort \t 8.Display\t");
            System.out.println("Enter the choice ");
            c=Integer.parseInt(in.readLine());
            switch(c)
            {
                case 1:
                {
                    System.out.println("Enter the string ");
                    str=in.readLine();
                    obj.add(str);
                    break;
                }
            }
        }
    }
}
```

```

case 2:
{
    System.out.println("Enter the string ");
    str=in.readLine();
    System.out.println("Specify the index/position to insert");
    i=Integer.parseInt(in.readLine());
    obj.add(i-1,str);
    System.out.println("The array list has following elements:"+obj);
    break;
}
case 3:
{
    System.out.println("Enter the string to search ");
    str=in.readLine();
    j=obj.indexOf(str);
    if(j==-1)
        System.out.println("Element not found");
    else
        System.out.println("Index of:"+str+"is"+j);
    break;
}
case 4:
{
    System.out.println("Enter the character to List string that starts with specified
character");
    str=in.readLine();
    for(i=0;i<(obj.size()-1);i++)
    {
        str1=obj.get(i);
        if(str1.startsWith(str))
        {
            System.out.println(str1);
        }
    }
    break;
}
case 5:
{
    System.out.println("Size of the list "+obj.size());
    break;
}
case 6:
{
    System.out.println("Enter the element to remove");
    str=in.readLine();
    if(obj.remove(str))
    {
        System.out.println("Element Removed"+str);
    }
    else
    {
        System.out.println("Element not present");
    }
    break;
}

```

```

        case 7:
    {
        Collections.sort(obj);
        System.out.println("The array list has following elements:"+obj);
        break;
    }
    case 8:
    {
        System.out.println("The array list has following elements:"+obj);
        break;
    }
}
System.out.println("Please Enter 0 to break and 1 to continue");
ch=Integer.parseInt(in.readLine());
}while(ch==1);
}
}

```

OUTPUT

D:\Java Programs>javac Arraylistexample.java
D:\Java Programs>java Arraylistexample

STRING MANIPULATION

- 1. Append at end 2.Insert at particular index 3.Search
- 4. List string that starting with letter
- 5. Size 6.Remove 7.Sort 8.Display

Enter the choice: 1

Enter the string: FIRST

Enter 0 to break and 1 to continue : 1

STRING MANIPULATION

- 1. Append at end 2.Insert at particular index 3.Search
- 4. List string that starting with letter
- 5. Size 6.Remove 7.Sort 8.Display

Enter the choice: 1

Enter the string : LAST

Enter 0 to break and 1 to continue : 1

STRING MANIPULATION

- 1. Append at end 2.Insert at particular index 3.Search
- 4. List string that starting with letter
- 5. Size 6.Remove 7.Sort 8.Display

Enter the choice : 8

The array list has following elements:[FIRST, LAST]

Enter 0 to break and 1 to continue : 1

STRING MANIPULATION

- 1. Append at end 2.Insert at particular index 3.Search
- 4. List string that starting with letter
- 5. Size 6.Remove 7.Sort 8.Display

Enter the choice : 2

Enter the string: SECOND

Specify the index/position to insert : 1

The array list has following elements:[SECOND, FIRST, LAST]

Enter 0 to break and 1 to continue: 1

STRING MANIPULATION

```
*****  
1. Append at end    2.Insert at particular index 3.Search  
4. List string that starting with letter  
5. Size      6.Remove     7.Sort      8.Display
```

Enter the choice : 3

Enter the string to search: LAST

Index of: LAST is 2

Enter 0 to break and 1 to continue : 1

STRING MANIPULATION

```
*****  
1. Append at end    2.Insert at particular index 3.Search  
4. List string that starting with letter  
5. Size      6.Remove     7.Sort      8.Display
```

Enter the choice : 5

Size of the list 3

Enter 0 to break and 1 to continue: 1

STRING MANIPULATION

```
*****  
1. Append at end    2.Insert at particular index 3.Search  
4. List string that starting with letter  
5. Size      6.Remove     7.Sort      8.Display
```

Enter the choice : 7

The array list has following elements:[FIRST, LAST, SECOND]

Enter 0 to break and 1 to continue : 0

RESULT

Thus the java program to perform string operations using ArrayList has been implemented and executed successfully.

AIM

To write a Java program to calculate the area of rectangle, circle and triangle using the concept of abstract class.

ALGORITHM:

1. Start
2. Create an abstract class named **shape** that contains two integers and an empty method named **printarea()**.
3. Provide three classes named rectangle, triangle and circle such that each one of the classes extends the class Shape.
4. Each of the inherited class from shape class should provide the implementation for the method **printarea()**.
5. Get the input and calculate the area of rectangle, circle and triangle.
6. In the **shapeclass**, create the objects for the three inherited classes and invoke the methods and display the area values of the different shapes.
7. Stop.

PROGRAM

Shapeclass.java

```
import java.util.*;
abstract class shape
{
    int a,b;
    abstract public void printarea();
}
class rectangle extends shape
{
    public int area_rect;
    public void printarea()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the length and breadth of rectangle");
        a=s.nextInt();
        b=s.nextInt();
        area_rect=a*b;
        System.out.println("Length of rectangle: "+a +"breadth of rectangle: "+b);
        System.out.println("The area of rectangle is:"+area_rect);
    }
}
class triangle extends shape
{
    double area_tri;
    public void printarea()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the base and height of triangle:");
        a=s.nextInt();
        b=s.nextInt();
        System.out.println("Base of triangle: "+a +"height of triangle: "+b);
        area_tri=(0.5*a*b);
        System.out.println("The area of triangle is:"+area_tri);
    }
}
```

```

class circle extends shape
{
    double area_circle;
    public void printarea()
    {
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the radius of circle:");
        a=s.nextInt();
        area_circle=(3.14*a*a);
        System.out.println("Radius of circle:"+a);
        System.out.println("The area of circle is:"+area_circle);
    }
}

public class Shapeclass
{
    public static void main(String[] args)
    {
        rectangle r=new rectangle();
        r.printarea();
        triangle t=new triangle();
        t.printarea();
        circle r1=new circle();
        r1.printarea();
    }
}

```

OUTPUT

D:\Java Programs>javac Shapeclass.java
D:\Java Programs>java Shapeclass
Enter the length and breadth of rectangle:
2
3
Length of rectangle: 2 breadth of rectangle: 3
The area of rectangle is:6

Enter the base and height of triangle:
5
6
Base of triangle: 5 height of triangle: 6
The area of triangle is: 15.0

Enter the radius of circle
4
Radius of circle: 4
The area of circle is:50.24

RESULT

Thus the Java program for calculate the area of rectangle, circle and triangle was implemented and executed successfully.

AIM

To write a Java program to implement user defined exception handling.

ALGORITHM:

1. Start
2. Create a class **NegativeAmtException** which extends Exception class.
3. Create a constructor which receives the string as argument.
4. Get the Amount as input from the user.
5. If the amount is negative, the exception will be generated.
6. Using the exception handling mechanism , the thrown exception is handled by the catch construct.
7. After the exception is handled , the string “invalid amount “ will be displayed.
8. If the amount is greater than 0, the message “Amount Deposited “ will be displayed
9. Stop.

PROGRAM 1:

userdefined.java

```
import java.util.*;
class NegativeAmtException extends Exception
{
    String msg;
    NegativeAmtException(String msg)
    {
        this.msg=msg;
    }
    public String toString()
    {
        return msg;
    }
}
public class userdefined
{
    public static void main(String[] args)
    {
        Scanner s=new Scanner(System.in);
        System.out.print("Enter Amount:");
        int a=s.nextInt();
        try
        {
            if(a<0)
            {
                throw new NegativeAmtException("Invalid Amount");
            }
            System.out.println("Amount Deposited");
        }
        catch(NegativeAmtException e)
        {
            System.out.println(e);
        }
    }
}
```

OUTPUT



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window contains the following text:

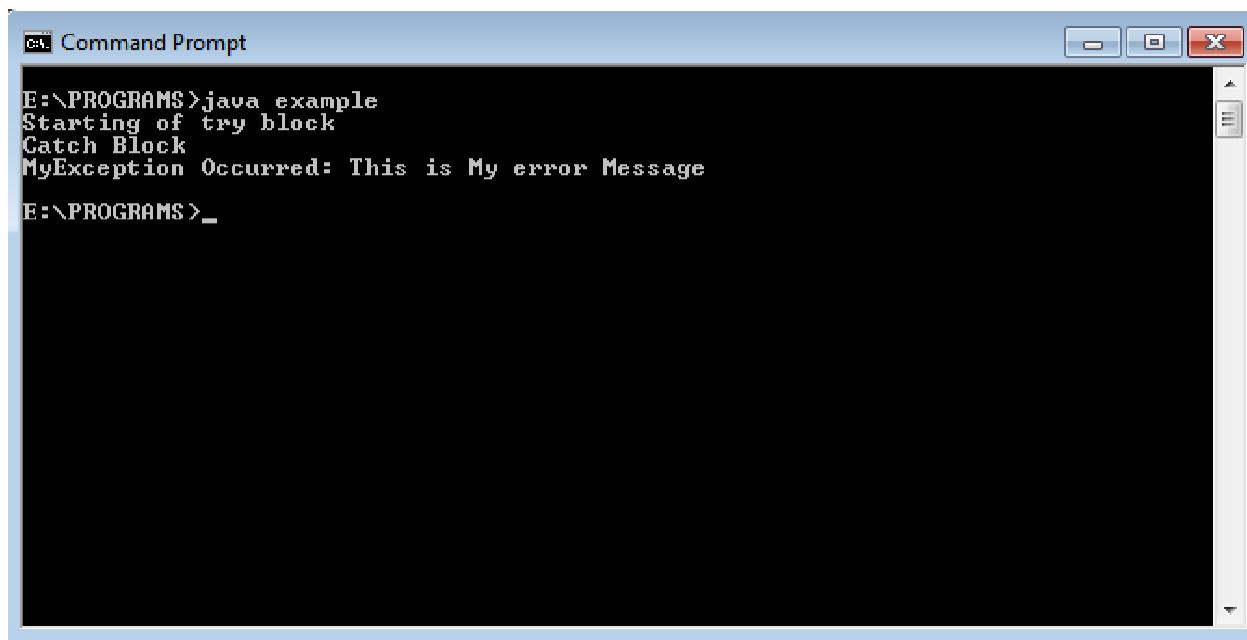
```
E:\>javac userdefined.java
E:\>java userdefined
Enter Amount:1000
Amount Deposited
E:\>java userdefined
Enter Amount:-1000
Invalid Amount
E:\>
```

PROGRAM 2:

example.java

```
class MyException extends Exception
{
    String str1;
    MyException(String str2)
    {
        str1=str2;
    }
    public String toString()
    {
        return ("MyException Occurred: "+str1);
    }
}
class example
{
    public static void main(String args[])
    {
        try
        {
            System.out.println("Starting of try block");
            throw new MyException("This is My error Message");
        }
        catch(MyException exp)
        {
            System.out.println("Catch Block");
            System.out.println(exp);
        }
    }
}
```

OUTPUT



The screenshot shows a Windows Command Prompt window titled "Command Prompt". The window contains the following text output:

```
E:\PROGRAMS>java example
Starting of try block
Catch Block
MyException Occurred: This is My error Message
E:\PROGRAMS>_
```

RESULT

Thus the Java program to implement user defined exception handling has been implemented and executed successfully.

AIM

To write a java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.

ALGORITHM:

1. Start
2. Create a class **filedemo**. Get the file name from the user.
3. Use the file functions and display the information about the file.
4. `getName()` displays the name of the file.
5. `getPath()` displays the path name of the file.
6. `getParent()` -This method returns the pathname string of this abstract pathname's parent, or
7. null if this pathname does not name a parent directory.
8. `exists()` – Checks whether the file exists or not.
9. `canRead()`-This method is basically a check if the file can be read.
10. `canWrite()`-verifies whether the application can write to the file.
11. `isDirectory()` – displays whether it is a directory or not.
12. `isFile()` – displays whether it is a file or not.
13. `lastmodified()` – displays the last modified information.
14. `length()`- displays the size of the file.
15. `delete()` – deletes the file
16. Invoke the predefined functions to display the information about the file.
17. Stop.

PROGRAM**filedemo.java**

```
import java.io.*;
import java.util.*;
class filedemo
{
    public static void main(String args[])
    {
        String filename;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter the file name ");
        filename=s.nextLine();
        File f1=new File(filename);
        System.out.println("*****");
        System.out.println("FILE INFORMATION");
        System.out.println("*****");
        System.out.println("NAME OF THE FILE "+f1.getName());
        System.out.println("PATH OF THE FILE "+f1.getPath());
        System.out.println("PARENT"+f1.getParent());
        if(f1.exists())
            System.out.println("THE FILE EXISTS ");
        else
            System.out.println("THE FILE DOES NOT EXISTS ");
        if(f1.canRead())
            System.out.println("THE FILE CAN BE READ ");
        else
            System.out.println("THE FILE CANNOT BE READ ");
        if(f1.canWrite())
            System.out.println("WRITE OPERATION IS PERMITTED");
```

```

        else
            System.out.println("WRITE OPERATION IS NOT PERMITTED");
        if(f1.isDirectory())
            System.out.println("IT IS A DIRECTORY ");
        else
            System.out.println("NOT A DIRECTORY");
        if(f1.isFile())
            System.out.println("IT IS A FILE ");
        else
            System.out.println("NOT A FILE");
        System.out.println("File last modified "+ f1.lastModified());
        System.out.println("LENGTH OF THE FILE "+f1.length());
        System.out.println("FILE DELETED "+f1.delete());
    }
}

```

OUTPUT

```

E:\>PROGRAMES>java filedemo
Enter the file name
teststack.java
*****
FILE INFORMATION
*****
NAME OF THE FILE teststack.java
PATH OF THE FILE teststack.java
PARENTnull
THE FILE EXISTS
THE FILE CAN BE READ
WRITE OPERATION IS PERMITTED
NOT A DIRECTORY
IT IS A FILE
File last modified 1528349052138
LENGTH OF THE FILE 1501
FILE DELETED true

E:\>PROGRAMES>_

```

RESULT

Thus the java program to display file information has been implemented and executed successfully.

AIM

To write a java program that implements a multi-threaded application.

ALGORITHM:

1. Start
2. Create a class **even** which implements first thread that computes the square of the number .
3. run() method implements the code to be executed when thread gets executed.
4. Create a class **odd** which implements second thread that computes the cube of the number.
5. Create a third thread that generates random number. If the random number is even, it displays the square of the number. If the random number generated is odd, it displays the cube of the given number.
6. The Multithreading is performed and the task switched between multiple threads.
7. The sleep () method makes the thread to suspend for the specified time.
8. Stop.

PROGRAM**multithreadprog.java**

```
import java.util.*;
class even implements Runnable
{
    public int x;
    public even(int x)
    {
        this.x = x;
    }
    public void run()
    {
        System.out.println("New Thread "+x +" is EVEN and Square of " + x + " is: " + x * x);
    }
}
class odd implements Runnable
{
    public int x;
    public odd(int x)
    {
        this.x = x;
    }
    public void run()
    {
        System.out.println("New Thread "+x +" is ODD and Cube of " + x + " is: " + x * x * x);
    }
}
class A extends Thread
{
    public void run()
    {
        int num = 0;
        Random r = new Random();
        try
        {
            for (int i = 0; i < 5; i++)
            {
                num = r.nextInt(100);
            }
        }
    }
}
```

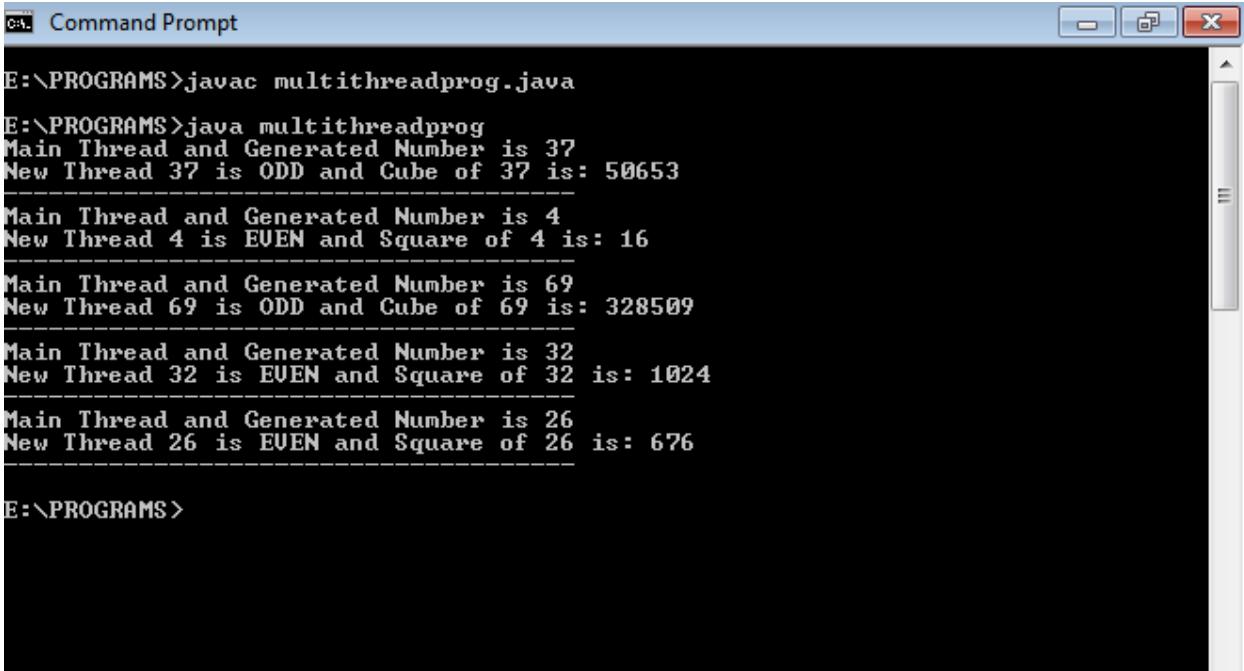
```

        System.out.println("Main Thread and Generated Number is " + num);
        if (num % 2 == 0)
        {
            Thread t1 = new Thread(new even(num));
            t1.start();
        }
        else
        {
            Thread t2 = new Thread(new odd(num));
            t2.start();
        }
        Thread.sleep(1000);
        System.out.println(".....");
    }
}
catch (Exception ex)
{
    System.out.println(ex.getMessage());
}
}

public class multithreadprog
{
    public static void main(String[] args)
    {
        A a = new A();
        a.start();
    }
}

```

OUTPUT



The screenshot shows a Windows Command Prompt window titled 'Command Prompt'. The command 'javac multithreadprog.java' is run, followed by 'java multithreadprog'. The output displays five pairs of generated numbers and their properties (odd/even and square/cube) from the main thread and two separate threads.

```

E:\>PROGRAMS>javac multithreadprog.java
E:\>PROGRAMS>java multithreadprog
Main Thread and Generated Number is 37
New Thread 37 is ODD and Cube of 37 is: 50653
-----
Main Thread and Generated Number is 4
New Thread 4 is EVEN and Square of 4 is: 16
-----
Main Thread and Generated Number is 69
New Thread 69 is ODD and Cube of 69 is: 328509
-----
Main Thread and Generated Number is 32
New Thread 32 is EVEN and Square of 32 is: 1024
-----
Main Thread and Generated Number is 26
New Thread 26 is EVEN and Square of 26 is: 676
-----
E:\>PROGRAMS>

```

RESULT

Thus the Java program for multi-threaded application has been implemented and executed successfully.

To write a java program to find the maximum value from the given type of elements using a generic function.

ALGORITHM:

1. Start
2. Create a class **Myclass** to implement generic class and generic methods.
3. Get the set of the values belonging to specific data type.
4. Create the objects of the class to hold integer, character and double values.
5. Create the method to compare the values and find the maximum value stored in the array.
6. Invoke the method with integer, character or double values. The output will be displayed based on the data type passed to the method.
7. Stop.

PROGRAM**genericdemo.java**

```
class MyClass<T extends Comparable<T>>
{
    T[] vals;
    MyClass(T[] o)
    {
        vals = o;
    }
    public T min()
    {
        T v = vals[0];
        for(int i=1; i < vals.length; i++)
            if(vals[i].compareTo(v) < 0)
                v = vals[i];
        return v;
    }
    public T max()
    {
        T v = vals[0];
        for(int i=1; i < vals.length;i++)
            if(vals[i].compareTo(v) > 0)
                v = vals[i];
        return v;
    }
}
class genericdemo
{
    public static void main(String args[])
    {
        int i;
        Integer inums[]={10,2,5,4,6,1};
        Character chs[]={ 'v','p','s','a','n','h'};
        Double d[]={20.2,45.4,71.6,88.3,54.6,10.4};
        MyClass<Integer> iob = new MyClass<Integer>(inums);
        MyClass<Character> cob = new MyClass<Character>(chs);
        MyClass<Double>dob = new MyClass<Double>(d);
        System.out.println("Max value in inums: " + iob.max());
        System.out.println("Min value in inums: " + iob.min());
        System.out.println("Max value in chs: " + cob.max());
        System.out.println("Min value in chs: " + cob.min());
        System.out.println("Max value in chs: " + dob.max());
        System.out.println("Min value in chs: " + dob.min());
    }
}
```

OUTPUT

```
D:\>Java Prgs>javac genericdemo.java
D:\>Java Prgs>java genericdemo
Max value in inums: 10
Max value in inums: 1
Max value in chs: v
Max value in chs: a
Max value in chs: 88.3
Max value in chs: 10.4
```

RESULT

Thus the Java program to find the maximum value from the given type of elements has been implemented using generics and executed successfully.

AIM

To design a calculator using event driven programming paradigm of Java with the following options

- a) Decimal Manipulations
- b) Scientific Manipulations

ALGORITHM:

1. Start
2. Import the swing packages and awt packages.
3. Create the class **scientificcalculator** that implements action listener.
4. Create the container and add controls for digits, scientific calculations and decimal Manipulations.
5. The different layouts can be used to lay the controls.
6. When the user presses the control, the event is generated and handled.
7. The corresponding decimal, numeric and scientific calculations are performed.
8. Stop

PROGRAM

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import javax.swing.event.*;
public class ScientificCalculator extends JFrame implements ActionListener
{
    JTextField tfield;
    double temp, temp1, result, a;
    static double m1, m2;
    int k = 1, x = 0, y = 0, z = 0;
    char ch;
    JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, zero, clr, pow2, pow3, exp,
    fac, plus, min, div, log, rec, mul, eq, addSub, dot, mr, mc, mp,
    mm, sqrt, sin, cos, tan;
    Container cont;
    JPanel textPanel, buttonpanel;
    ScientificCalculator()
    {
        cont = getContentPane();
        cont.setLayout(new BorderLayout());
        JPanel textpanel = new JPanel();
        tfield = new JTextField(25);
        tfield.setHorizontalAlignment(SwingConstants.RIGHT);
        tfield.addKeyListener(new KeyAdapter()
        {
            public void keyTyped(KeyEvent keyevent)
            {
                char c = keyevent.getKeyChar();
                if (c >= '0' && c <= '9') { }
                else
                {
                    keyevent.consume();
                }
            }
        });
        textpanel.add(tfield);
```

```
buttonpanel = new JPanel();
buttonpanel.setLayout(new GridLayout(8, 4, 2, 2));
boolean t = true;
mr = new JButton("MR");
buttonpanel.add(mr);
mr.addActionListener(this);
mc = new JButton("MC");
buttonpanel.add(mc);
mc.addActionListener(this);
mp = new JButton("M+");
buttonpanel.add(mp);
mp.addActionListener(this);
mm = new JButton("M-");
buttonpanel.add(mm);
mm.addActionListener(this);
b1 = new JButton("1");
buttonpanel.add(b1);
b1.addActionListener(this);
b2 = new JButton("2");
buttonpanel.add(b2);
b2.addActionListener(this);
b3 = new JButton("3");
buttonpanel.add(b3);
b3.addActionListener(this);
b4 = new JButton("4");
buttonpanel.add(b4);
b4.addActionListener(this);
b5 = new JButton("5");
buttonpanel.add(b5);
b5.addActionListener(this);
b6 = new JButton("6");
buttonpanel.add(b6);
b6.addActionListener(this);
b7 = new JButton("7");
buttonpanel.add(b7);
b7.addActionListener(this);
b8 = new JButton("8");
buttonpanel.add(b8);
b8.addActionListener(this);
b9 = new JButton("9");
buttonpanel.add(b9);
b9.addActionListener(this);
zero = new JButton("0");
buttonpanel.add(zero);
zero.addActionListener(this);
plus = new JButton("+");
buttonpanel.add(plus);
plus.addActionListener(this);
min = new JButton("-");
buttonpanel.add(min);
min.addActionListener(this);
mul = new JButton("*");
buttonpanel.add(mul);
mul.addActionListener(this);
div = new JButton("/");
div.addActionListener(this);
buttonpanel.add(div);
addSub = new JButton("+/-");
buttonpanel.add(addSub);
```

```

addSub.addActionListener(this);
dot = new JButton(".");
buttonpanel.add(dot);
dot.addActionListener(this);
eq = new JButton("=");
buttonpanel.add(eq);
eq.addActionListener(this);
rec = new JButton("1/x");
buttonpanel.add(rec);
rec.addActionListener(this);
sqrt = new JButton("Sqrt");
buttonpanel.add(sqrt);
sqrt.addActionListener(this);
log = new JButton("log");
buttonpanel.add(log);
log.addActionListener(this);
sin = new JButton("SIN");
buttonpanel.add(sin);
sin.addActionListener(this);
cos = new JButton("COS");
buttonpanel.add(cos);
cos.addActionListener(this);
tan = new JButton("TAN");
buttonpanel.add(tan);
tan.addActionListener(this);
pow2 = new JButton("x^2");
buttonpanel.add(pow2);
pow2.addActionListener(this);
pow3 = new JButton("x^3");
buttonpanel.add(pow3);
pow3.addActionListener(this);
exp = new JButton("Exp");
exp.addActionListener(this);
buttonpanel.add(exp);
fac = new JButton("n!");
fac.addActionListener(this);
buttonpanel.add(fac);
clr = new JButton("AC");
buttonpanel.add(clr);
clr.addActionListener(this);
cont.add("Center", buttonpanel);
cont.add("North", textpanel);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public void actionPerformed(ActionEvent e)
{
    String s = e.getActionCommand();
    if (s.equals("1"))
    {
        if (z == 0)
        {
            tfield.setText(tfield.getText() + "1");
        }
        else {
            tfield.setText("");
            tfield.setText(tfield.getText() + "1");
            z = 0;
        }
    }
}

```

```

if (s.equals("2")) {
if (z == 0) {
tfield.setText(tfield.getText() + "2");
}
else
{
tfield.setText("");
tfield.setText(tfield.getText() + "2");
z = 0;
}
}
if (s.equals("3")) {
if (z == 0) {
tfield.setText(tfield.getText() + "3");
}
else
{
tfield.setText("");
tfield.setText(tfield.getText() + "3");
z = 0;
}
}
if (s.equals("4")) {
if (z == 0) {
tfield.setText(tfield.getText() + "4");
}
else
{
tfield.setText("");
tfield.setText(tfield.getText() + "4");
z = 0;
}
}
if (s.equals("5")) {
if (z == 0) {
tfield.setText(tfield.getText() + "5");
}
else
{
tfield.setText("");
tfield.setText(tfield.getText() + "5");
z = 0;
}
}
if (s.equals("6")) {
if (z == 0) {
tfield.setText(tfield.getText() + "6");
}
else
{
tfield.setText("");
tfield.setText(tfield.getText() + "6");
z = 0;
}
}
if (s.equals("7")) {
if (z == 0) {
tfield.setText(tfield.getText() + "7");
}
}

```

```

else
{
tfield.setText("");
tfield.setText(tfield.getText() + "7");
z = 0;
}
}
if (s.equals("8")) {
if (z == 0) {
tfield.setText(tfield.getText() + "8");
}
else
{
tfield.setText("");
tfield.setText(tfield.getText() + "8");
z = 0;
}
}
if (s.equals("9")) {
if (z == 0) {
tfield.setText(tfield.getText() + "9");
}
else
{
tfield.setText("");
tfield.setText(tfield.getText() + "9");
z = 0;
}
}
if (s.equals("0"))
{
if (z == 0) {
tfield.setText(tfield.getText() + "0");
}
else
{
tfield.setText("");
tfield.setText(tfield.getText() + "0");
z = 0;
}
}
if (s.equals("AC")) {
tfield.setText("");
x = 0;
y = 0;
z = 0;
}
if (s.equals("log"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.log(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}

```

```

if (s.equals("1/x")) {
if (tfield.getText().equals("")) {
tfield.setText("");
}
else
{
a = 1 / Double.parseDouble(tfield.getText());
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("Exp")) {
if (tfield.getText().equals("")) {
tfield.setText("");
}
else
{
a = Math.exp(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("x^2")) {
if (tfield.getText().equals("")) {
tfield.setText("");
}
else
{
a = Math.pow(Double.parseDouble(tfield.getText()), 2);
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("x^3")) {
if (tfield.getText().equals("")) {
tfield.setText("");
}
else
{
a = Math.pow(Double.parseDouble(tfield.getText()), 3);
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("+/-")) {
if (x == 0) {
tfield.setText("-" + tfield.getText());
x = 1;
}
else
{
tfield.setText(tfield.getText());
}
}
if (s.equals(".")) {
if (y == 0) {
tfield.setText(tfield.getText() + ".");
y = 1;
}
}

```

```

else
{
tfield.setText(tfield.getText());
}
}
if (s.equals("+"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
temp = 0;
ch = '+';
}
else
{
temp = Double.parseDouble(tfield.getText());
tfield.setText("");
ch = '+';
y = 0;
x = 0;
}
tfield.requestFocus();
}
if (s.equals("-"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
temp = 0;
ch = '-';
}
else
{
x = 0;
y = 0;
temp = Double.parseDouble(tfield.getText());
tfield.setText("");
ch = '-';
}
tfield.requestFocus();
}
if (s.equals("/")) {
if (tfield.getText().equals(""))
{
tfield.setText("");
temp = 1;
ch = '/';
}
else
{
x = 0;
y = 0;
temp = Double.parseDouble(tfield.getText());
ch = '/';
}
tfield.setText("");
}
tfield.requestFocus();
}
if (s.equals("*")) {

```

```

if(tfield.getText().equals(""))
{
tfield.setText("");
temp = 1;
ch = '*';
}
else
{
x = 0;
y = 0;
temp = Double.parseDouble(tfield.getText());
ch = '*';
tfield.setText("");
}
tfield.requestFocus();
}
if(s.equals("MC"))
{
m1 = 0;
tfield.setText("");
}
if(s.equals("MR"))
{
tfield.setText("");
tfield.setText(tfield.getText() + m1);
}
if(s.equals("M+"))
{
if(k == 1)
{
m1 = Double.parseDouble(tfield.getText());
k++;
}
else
{
m1 += Double.parseDouble(tfield.getText());
tfield.setText("") + m1;
}
}
if(s.equals("M-"))
{
if(k == 1)
{
m1 = Double.parseDouble(tfield.getText());
k++;
}
else
{
m1 -= Double.parseDouble(tfield.getText());
tfield.setText("") + m1;
}
}
if(s.equals("Sqrt"))
{
if(tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.sqrt(Double.parseDouble(tfield.getText()));
}
}

```

```

tfield.setText("");
field.setText(tfield.getText() + a);
}
}
if (s.equals("SIN"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.sin(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("COS"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.cos(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("TAN")) {
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = Math.tan(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
if (s.equals("="))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
temp1 = Double.parseDouble(tfield.getText());
switch (ch)
{
case '+':
result = temp + temp1;
break;
case '-':
result = temp - temp1;
break;
case '/':

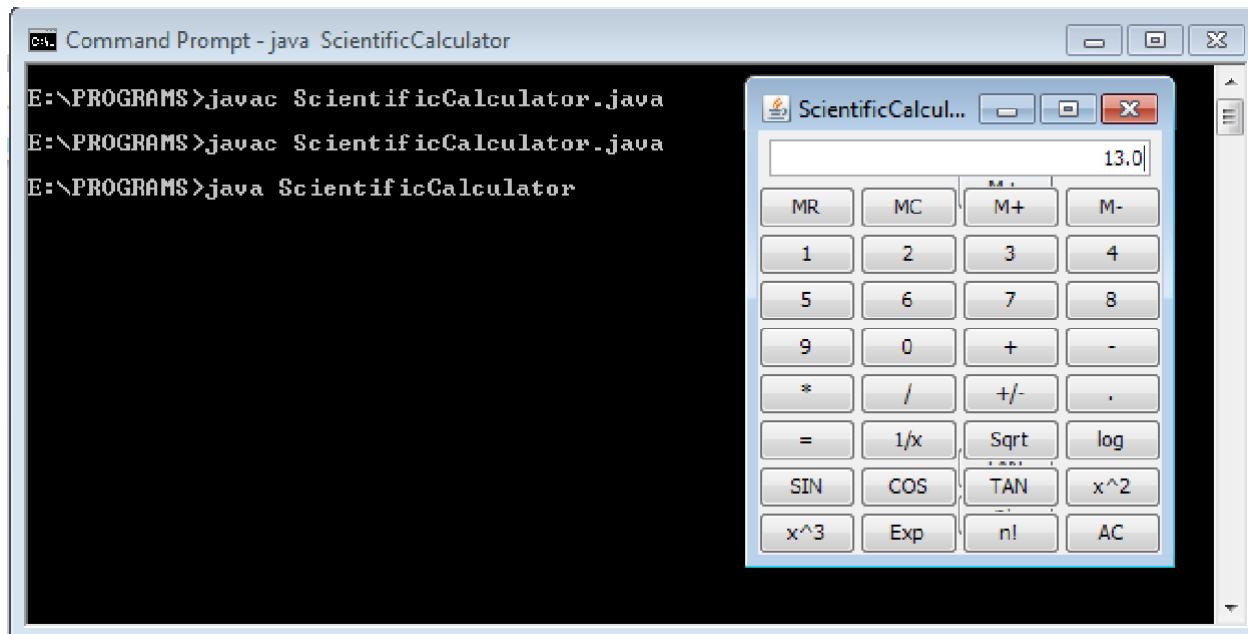
```

```

result = temp / temp1;
break;
case '*':
result = temp * temp1;
break;
}
tfield.setText("");
tfield.setText(tfield.getText() + result);
z = 1;
}
}
if (s.equals("n!"))
{
if (tfield.getText().equals(""))
{
tfield.setText("");
}
else
{
a = fact(Double.parseDouble(tfield.getText()));
tfield.setText("");
tfield.setText(tfield.getText() + a);
}
}
tfield.requestFocus();
}
double fact(double x)
{
    int er = 0;
    if (x < 0)
    {
        er = 20;
        return 0;
    }
    double i, s = 1;
    for (i = 2; i <= x; i += 1.0)
        s *= i;
    return s;
}
public static void main(String args[])
{
    try
    {
        UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
    }
    catch (Exception e)
    {
    }
    ScientificCalculator f = new ScientificCalculator();
    f.setTitle("ScientificCalculator");
    f.pack();
    f.setVisible(true);
}
}

```

OUTPUT



RESULT

Thus the Java programs for scientific calculator has been implemented and executed successfully.