



GEETHANJALI INSTITUTE OF SCIENCE & TECHNOLOGY

(Approved by AICTE, New Delhi & Affiliated to JNTU, Anantapur, Accredited by NAAC-'A' grade)

Bombay Highway, Gangavaram (V), Kovur(M), SPSR Nellore (Dt), Andhra Pradesh, India- 524137

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

Lab Manual

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
II-YEAR-I-SEMESTER
R-20 REGULATION
ACADEMIC YEAR: 2021-2022**

Institute Vision and Mission

Vision

- To emerge as a leading Engineering institution imparting quality education

Mission

- IM₁ Implement Effective teaching-learning strategies for quality education
- IM₂ Build Congenial academic ambience for progressive learning
- IM₃ Facilitate Skill development through Industry-Institute initiatives
- IM₄ Groom environmentally conscious and socially responsible technocrats

Department of Computer Science and Engineering

Vision

- To develop as a lead learning resource centre producing skilled professionals

Mission

- DM₁ Provide dynamic and application oriented education through advanced teaching learning methodologies
- DM₂ Create sufficient physical infrastructural facilities to enhance learning
- DM₃ Strengthen the professional skills through effective Industry- Institute Interaction
- DM₄ Organize personality development activities to inculcate life skills and ethical values


Program Specific Outcomes (PSOs)

- **PSO1 Professional Knowledge:** Analyse and apply the concepts of Algorithms, Web Technologies and Data Analytics to meet specified requirements.
- **PSO2 Software Skills:** Design and implement solutions for computing problems using Java, PHP, Python and Big Data technologies.

Program Educational Objectives (PEOs)

Graduates of B.Tech in Computer Science and Engineering Programme shall be able to

- **PEO1** Develop expertise in logical reasoning, analysis and design to solve Computer Science and Engineering problems.
- **PEO2** Competent to work as an individual or team member contributing to research and solve real world problems.
- **PEO3** Involve in multi disciplinary teams by imparting interpersonal skills and ethical behaviour.
- **PEO4** Engage in Life Long Learning for career enhancement and professional growth

	GEETHANJALI INSTITUTE OF SCIENCE & TECHNOLOGY
	(Approved by AICTE, New Delhi & Affiliated to JNTU, Anantapur, Accredited by NAAC- 'A' grade)
	Bombay Highway, Gangavaram (V), Kovur(M), SPSR Nellore (Dt), Andhra Pradesh, India- 524137

COURSE	
BRANCH	
REGULATION	
YEAR AND SEMESTER	
COURSE CODE	
SUBJECT	
NAME OF THE STUDENT	
UNIVERSITY ROLL NO	
SECTION	
ACADEMIC YEAR	

GEETHANJALI INSTITUTE OF SCIENCE AND TECHNOLOGY-NELLORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

REGULATION-R20-II-YEAR-I-SEMESTER

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

LIST OF PROGRAMS			
Program No	Title of the Program	Date	Signature of the Faculty
1	Finding Prime numbers between 1 to n		
2	Printing all the real solutions of the Quadratic equation		
3	Generating Electricity Bill		
4	Finding the Product of two given matrices		
5	Implementing Single Inheritance		
6	Illustrating the usage of final keyword		
7	Illustrating Abstract class		
8	Illustrating Dynamic Binding		
9	Illustrating Method Overloading		
10	Illustrating Method Overriding		
11	Currency Conversion using Interfaces		
12	Displaying unique values from the given set of numbers		
13	Illustrating the usage of StringTokenizer class		
14	Creating a User defined Exception		
15	Performing Division of two numbers using Exceptions		
16	Executing multiple threads		
17	Splitting a given text file into n parts		
18	Displaying the information about a given file		
19	Displaying the number of characters, words and lines in a text file		
20	Displaying the contents of a file along with line number		

GEETHANJALI INSTITUTE OF SCIENCE AND TECHNOLOGY-NELLORE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

REGULATION-R20-II-YEAR-I-SEMESTER

OBJECT ORIENTED PROGRAMMING THROUGH JAVA

LIST OF PROGRAMS			
Program No	Title of the Program	Date	Signature of the Faculty
21	Implementing Producer-Consumer Problem using threads		
22	Implementing a Stack using an array		
23	Performing Arithmetic operations using user interface components		
24	Simulating a traffic light		
25	Simulating a waving flag		
26	Conversion of Decimal to Binary using User Interface Components		
27	Handling Mouse Events		
28	Handling Key Events		
29	Finding the maximum of the given elements using a generic function		
30	JDBC application with insert and view options		
ADDITIONAL PROGRAMS			
1	Generation of N-terms of Fibonacci Series using recursion		
2	Implementing Multiple Inheritance		

1. Finding Prime numbers between 1 and n

Aim: To develop a java program for printing prime numbers between 1 and n

Program:

//JAVA PROGRAM TO PRINT PRIME NUMBERS FROM 1 TO N

```
import java.util.*;

class Primegen
{
    public static void main(String args[])throws Exception
    {
        int n,i,j,fc;

        Scanner sc = new Scanner(System.in);

        System.out.print("\n\n\t ENTER THE VALUE FOR N....");

        n=sc.nextInt();

        System.out.print("\n\n\t THE PRIME NUMBERS BETWEEN 1 AND " + n + " ARE...");

        for(i=1;i<=n;i=i+1)
        {
            for(j=1,fc=0;j<=i;j=j+1)
            {
                if(i%j==0)
                {
                    fc=fc+1;
                }
            }

            if(fc==2)
            {
                System.out.print(" " +i);
```

```
}  
}  
}  
}
```

Input/Output:

ENTER THE VALUE FOR N....10

THE PRIME NUMBERS BETWEEN 1 AND 10 ARE... 2 3 5 7

2. Printing all the real solutions of the Quadratic equation

Aim: To develop a java program for printing the roots of the given quadratic equation.

Program:

```
// JAVA PROGRAM TO PRINT THE ROOTS OF A QUADRATIC EQUATION
```

```
import java.util.*;
```

```
class Quadratic
```

```
{
```

```
    public static void main(String args[])throws Exception
```

```
    {
```

```
        int a,b,c;
```

```
        double d,r,r1,r2,p,q,z;
```

```
        Scanner sc=new Scanner(System.in);
```

```
        System.out.print("\n\n\t ENTER THE COEFFICIENT OF X*X...");
```

```
        a=sc.nextInt();
```

```
        System.out.print("\n\n\t ENTER THE COEFFICIENT OF X...");
```

```
        b=sc.nextInt();
```

```
        System.out.print("\n\n\t ENTER THE CONSTANT TERM.....");
```

```
        c=sc.nextInt();
```

```
        d=(b*b)-(4*a*c);
```

```
        if(d==0)
```

```
        {
```

```
            r=-(b/(2*a));
```

```
            System.out.print("\n\n\t THE ROOTS ARE REAL AND EQUAL");
```

```
            System.out.print("\n\n\t THE ROOT IS...."+r);
```

```
        }
```



```

else if(d>0)
{
    p=-(b/(2*a));
    q=(Math.sqrt(d))/(2*a);
    r1=p+q;
    r2=p-q;
    System.out.print("\n\n\t THE ROOTS ARE REAL AND DISTINCT");
    System.out.print("\n\n\t THE ROOTS ARE...."+r1 +"AND"+r2);
}
else
{
    z=Math.abs(d);
    p=-(b/(2*a));
    q=(Math.sqrt(z))/(2*a);
    System.out.print("\n\n\t THE ROOTS ARE IMAGINARY");
    System.out.print("\n\n\t THE FIRST ROOT IS...."+p+"+"+q+"i");
    System.out.print("\n\n\t THE SECOND ROOT IS..." +p+"-"+q+"i");
}
}
}

```

Input/Output:

Trial Run-1:

ENTER THE COEFFICIENT OF X*X...2

ENTER THE COEFFICIENT OF X...3

ENTER THE CONSTANT TERM.....1

THE ROOTS ARE REAL AND DISTINCT

THE ROOTS ARE....0.25AND-0.25

Trial Run-2

ENTER THE COEFFICIENT OF X^2 ...2

ENTER THE COEFFICIENT OF X ...4

ENTER THE CONSTANT TERM.....2

THE ROOTS ARE REAL AND EQUAL

THE ROOT IS....-1.0

Trial Run-3

ENTER THE COEFFICIENT OF X^2 ...1

ENTER THE COEFFICIENT OF X ...-3

ENTER THE CONSTANT TERM.....9

THE ROOTS ARE IMAGINARY

THE FIRST ROOT IS.... $1.0+2.598076211353316i$

THE SECOND ROOT IS... $1.0-2.598076211353316i$

3. Generating Electricity Bill

Aim: To develop a java program for generating Electricity bill

Program:

```
// JAVA PROGRAM TO GENERATE ELECTRICITY BILL
```

```
import java.util.*;
```

```
class Ebill
```

```
{
```

```
    int cno;
```

```
    String cname;
```

```
    String ctype;
```

```
    int pmr,cmr,cu;
```

```
    double amt;
```

```
    public void accept( )throws Exception
```

```
    {
```

```
        Scanner sc= new Scanner(System.in);
```

```
        System.out.print("\n\n\t ENTER THE CONSUMER NUMBER...");
```

```
        cno=sc.nextInt( );
```

```
        System.out.print("\n\n\t ENTER THE CONSUMER NAME.....");
```

```
        cname=sc.next( );
```

```
        System.out.print("\n\n\t ENTER THE CONSUMER TYPE(DOMESTIC OR  
COMMERCIAL).....");
```

```
        ctype=sc.next( );
```

```
        System.out.print("\n\n\t ENTER THE PREVIOUS METER READING....");
```

```
        pmr=sc.nextInt( );
```

```
        System.out.print("\n\n\t ENTER THE CURRENT METER READING....");
```

```
        cmr=sc.nextInt( );
```

```

}

public void display( )
{
    System.out.print("\n\n\t-----");
    System.out.print("\n\n\t CONSUMER NO-----"+cno);
    System.out.print("\n\n\tCONSUMER NAME----"+cname);
    System.out.print("\n\n\tCONSUMER TYPE-----"+ctype);
    System.out.print("\n\n\t PREVIOUS METER READING-----"+pmr);
    System.out.print("\n\n\t CURRENT METER READING-----"+cmr);
    System.out.print("\n\n\t NO OF UNITS CONSUMED-----"+cu);
    System.out.print("\n\n\t TOTAL AMOUNT-----"+amt);
    System.out.print("\n\n\t-----");
}

public void compute( )
{
    cu=cmr-pmr;
    if(ctype.equals("DOMESTIC"))
    {
        if(cu<=100)
        {
            amt=cu*1.00;
        }
        else if(cu>100 && cu<=200)
        {
            amt=(100*1.00)+(cu-100)*2.50;
        }
        else if(cu>200 && cu<=500)

```

```
{  
    amt=(100*1.00)+(100*2.50)+(cu-200)*4.00;  
}  
else  
{  
    amt=(100*1.00)+(100*2.50)+(200*4.00)+(cu-500)*6;  
}  
  
}  
else if(ctype.equals("COMMERCIAL"))  
{  
  
    if(cu<=100)  
    {  
        amt=cu*2.00;  
    }  
    else if(cu>100 && cu<=200)  
    {  
        amt=(100*2.00)+(cu-100)*4.50;  
    }  
    else if(cu>200 && cu<=500)  
    {  
        amt=(100*2.00)+(100*4.50)+(cu-200)*6.00;  
    }  
    else  
    {  
        amt=(100*2.00)+(100*4.50)+(200*6.00)+(cu-500)*7;
```

```

    }

}

else

{

    System.out.print("\n\n\t INVALID CONSUMER TYPE");

}

}

}

class Edemo

{

    public static void main(String args[ ])throws Exception

    {

        Ebill eb=new Ebill( );

        eb.accept( );

        eb.compute( );

        eb.display( );

    }

}

```

Input/Output:

ENTER THE CONSUMER NUMBER...101

ENTER THE CONSUMER NAME.....KIRAN

ENTER THE CONSUMER TYPE(DOMESTIC OR COMMERCIAL).....DOMESTIC

ENTER THE PREVIOUS METER READING....100

ENTER THE CURRENT METER READING....290

CONSUMER NO-----101

CONSUMER NAME----KIRAN

CONSUMER TYPE-----DOMESTIC

PREVIOUS METER READING-----100

CURRENT METER READING-----290

NO OF UNITS CONSUMED-----190

TOTAL AMOUNT-----325.0

4. Finding the Product of two given matrices

Aim: To develop a java program for performing matrix multiplication

Program:

// JAVA PROGRAM TO PERFORM MATRIX MULTIPLICATION

```
import java.util.*;

class Matrixmul

{

    public static void main(String args[])throws Exception

    {

        int m,n,p,q,i,j,k,a[],b[],c[];

        Scanner sc= new Scanner(System.in);

        System.out.print("\n\n\t ENTER THE NUMBER OF ROWS IN THE FIRST MATRIX...");

        m=sc.nextInt();

        System.out.print("\n\n\t ENTER THE NUMBER OF COLUMNS IN THE FIRST MATRIX...");

        n=sc.nextInt();

        System.out.print("\n\n\t ENTER THE NUMBER OF ROWS IN THE SECOND MATRIX...");

        p=sc.nextInt();

        System.out.print("\n\n\t ENTER THE NUMBER OF COLUMNS IN THE SECOND MATRIX...");

        q=sc.nextInt();

        a=new int[m][n];

        b=new int[p][q];

        c=new int[m][q];

        if(n==p)

        {

            System.out.print("\n\n\t ENTER THE ELEMENTS OF THE FIRST MATRIX...");
```



```
for(i=0;i<m;i=i+1)
{
    for(j=0;j<n;j=j+1)
    {
        System.out.print("\n\n\t ENTER THE ELEMENT-a["+i+"]["+j+"]...");
        a[i][j]=sc.nextInt();
    }
}

System.out.print("\n\n\t ENTER THE ELEMENTS OF THE SECOND MATRIX...");
for(i=0;i<p;i=i+1)
{
    for(j=0;j<q;j=j+1)
    {
        System.out.print("\n\n\t ENTER THE ELEMENT-b["+i+"]["+j+"]...");
        b[i][j]=sc.nextInt();
    }
}

System.out.print("\n\n\t THE ELEMENTS OF THE FIRST MATRIX ARE...\n");
for(i=0;i<m;i=i+1)
{
    for(j=0;j<n;j=j+1)
    {
        System.out.print(" "+a[i][j]);
    }
    System.out.print("\n");
}

System.out.print("\n\n\t THE ELEMENTS OF THE SECOND MATRIX ARE...\n");
```

```

for(i=0;i<p;i=i+1)
{
    for(j=0;j<n;j=j+1)
    {
        System.out.print(" "+b[i][j]);
    }
    System.out.print("\n");
}
for(i=0;i<m;i=i+1)
{
    for(j=0;j<q;j=j+1)
    {
        c[i][j]=0;
    }
}
for(i=0;i<m;i=i+1)
{
    for(j=0;j<q;j=j+1)
    {
        for(k=0;k<n;k=k+1)
        {
            c[i][j]=c[i][j]+(a[i][k]*b[k][j]);
        }
    }
}

System.out.print("\n\n\t THE PRODUCT OF TWO MATRICES IS...\n");

for(i=0;i<m;i=i+1)

```

```

    {
        for(j=0;j<q;j=j+1)
        {
            System.out.print(" "+c[i][j]);
        }
        System.out.print("\n");
    }
}
else
{
    System.out.print("\n\n\t MATRIX MULTIPLICATION IS NOT POSSIBLE DUE TO
DIMENSIONS");
}
}
}

```

Input/Output:

Trial Run-1

ENTER THE NUMBER OF ROWS IN THE FIRST MATRIX...2

ENTER THE NUMBER OF COLUMNS IN THE FIRST MATRIX...2

ENTER THE NUMBER OF ROWS IN THE SECOND MATRIX...3

ENTER THE NUMBER OF COLUMNS IN THE SECOND MATRIX...3

MATRIX MULTIPLICATION IS NOT POSSIBLE DUE TO DIMENSIONS

Trial Run-2

ENTER THE NUMBER OF ROWS IN THE FIRST MATRIX...2

ENTER THE NUMBER OF COLUMNS IN THE FIRST MATRIX...2

ENTER THE NUMBER OF ROWS IN THE SECOND MATRIX...2

ENTER THE NUMBER OF COLUMNS IN THE SECOND MATRIX...2

ENTER THE ELEMENTS OF THE FIRST MATRIX...

ENTER THE ELEMENT-a[0][0]...1

ENTER THE ELEMENT-a[0][1]...2

ENTER THE ELEMENT-a[1][0]...3

ENTER THE ELEMENT-a[1][1]...4

ENTER THE ELEMENTS OF THE SECOND MATRIX...

ENTER THE ELEMENT-b[0][0]...1

ENTER THE ELEMENT-b[0][1]...0

ENTER THE ELEMENT-b[1][0]...0

ENTER THE ELEMENT-b[1][1]...1

THE ELEMENTS OF THE FIRST MATRIX ARE...

1 2

3 4

THE ELEMENTS OF THE SECOND MATRIX ARE...

1 0

0 1

THE PRODUCT OF TWO MATRICES IS...

1 2

3 4

5. Implementing Single Inheritance

Aim: To develop a java program to implement Single Inheritance

Program:

// JAVA PROGRAM TO ILLUSTRATE THE USE OF INHERITANCE

```
import java.util.*;
```

```
class Student
```

```
{
```

```
    int rollno;
```

```
    String name;
```

```
    public void accept_details( )throws Exception
```

```
    {
```

```
        Scanner sc= new Scanner(System.in);
```

```
        System.out.print("\n\n\t ENTER THE ROLLNO....");
```

```
        rollno=sc.nextInt( );
```

```
        System.out.print("\n\n\t ENTER THE NAME.....");
```

```
        name=sc.next( );
```

```
    }
```

```
    public void display_details( )
```

```
    {
```

```
        System.out.print("\n\n\t ROLLNO IS....."+rollno);
```

```
        System.out.print("\n\n\t NAME IS....."+name);
```

```
    }
```

```
}
```

```
class Test extends Student
```

```
{
```

```
    int m,p,c;
```

```

public void accept_marks( )throws Exception
{
    Scanner sc= new Scanner(System.in);

    System.out.print("\n\n\t ENTER THE MARKS IN MATHS.....");

    m=sc.nextInt( );

    System.out.print("\n\n\t ENTER THE MARKS IN PHYSICS....");

    p=sc.nextInt( );

    System.out.print("\n\n\t ENTER THE MARKS IN CHEMISTRY....");

    c=sc.nextInt( );

}

public void display_marks( )throws Exception
{
    System.out.print("\n\n\t MARKS IN MATHS....."+m);

    System.out.print("\n\n\t MARKS IN PHYSICS...."+p);

    System.out.print("\n\n\t MARKS IN CHEMISTRY...."+c);

}

}

class TestDemo
{
    public static void main(String args[ ])throws Exception
    {
        Test t= new Test( );

        t.accept_details( );

        t.accept_marks( );

        t.display_details( );

        t.display_marks( );

    }
}

```

}

Input/Output

ENTER THE ROLLNO....1000

ENTER THE NAME.....ravi

ENTER THE MARKS IN MATHS.....75

ENTER THE MARKS IN PHYSICS....75

ENTER THE MARKS IN CHEMISTRY....75

ROLLNO IS.....1000

NAME IS.....ravi

MARKS IN MATHS.....75

MARKS IN PHYSICS....75

MARKS IN CHEMISTRY....75

6. Illustrating the usage of final keyword

Aim: To develop a java program to illustrate the usage of final keyword with variable

Program:

// JAVA PROGRAM TO ILLUSTRATE THE USAGE OF FINAL KEYWORD WITH VARIABLES

```
class Finaldemo1
{
    public static void main(String args[])
    {
        final int n=10;
        System.out.print("\n\n\t THE VALUE OF n IS...."+n);
        n=15;
        System.out.print("\n\n\t THE VALUE OF n IS..." +n);
    }
}
```

Input/Output:

Finaldemo1.java:8: error: cannot assign a value to final variable n

```
n=15;
```

```
^
```

1 error

Aim: To develop a java program to illustrate the usage of final keyword with a method

Program:

//JAVA PROGRAM TO ILLUSTRATE THE USAGE OF FINAL KEYWORD WITH A METHOD

```
class Simple1
{
    public final void display( )
    {
        System.out.print("\n\n\t BASE CLASS");
    }
}

class Simple2 extends Simple1
{
    public void display( )
    {
        System.out.print("\n\n\t DERIVED CLASS");
    }
}

class Finaldemo2
{
    public static void main(String args[ ])
    {
        Simple2 s=new Simple2( );
        s.display( );
    }
}
```

Input/Output:

Finaldemo2.java:11: error: display() in Simple2 cannot override display() in Simple1

```
public void display( )
```

```
    ^
```

overridden method is final

1 error

Aim: To develop a java program for illustrating the usage of final keyword with class

Program:

//JAVA PROGRAM TO ILLUSTRATE THE USAGE OF FINAL KEYWORD WITH A CLASS

```
final class Simple1
{
    public void display( )
    {
        System.out.print("\n\n\t BASE CLASS");
    }
}

class Simple2 extends Simple1
{
    public void display( )
    {
        System.out.print("\n\n\t DERIVED CLASS");
    }
}

class Finaldemo3
{
    public static void main(String args[ ])
    {
        Simple2 s=new Simple2( );
        s.display( );
    }
}
```

```
}
```

Input/Output:

Finaldemo3.java:9: error: cannot inherit from final Simple1

class Simple2 extends Simple1

^

1 error

7. Illustrating Abstract class

Aim: To develop a java program for creating an abstract class and extend classes from it

Program:

//JAVA PROGRAM TO CREATE AN ABSTRACT CLASS AND EXTEND CLASSES FROM IT

```
import java.util.*;
```

```
abstract class Shape
```

```
{
```

```
    int x,y;
```

```
    abstract public void area( );
```

```
}
```

```
class Rectangle extends Shape
```

```
{
```

```
    public void accept( ) throws Exception
```

```
    {
```

```
        Scanner sc= new Scanner(System.in);
```

```
        System.out.print("\n\n\t ENTER THE LENGTH OF THE RECTANGLE...");
```

```
        x=sc.nextInt( );
```

```
        System.out.print("\n\n\t ENTER THE BREADTH OF THE RECTANGLE....");
```

```
        y=sc.nextInt( );
```

```
    }
```

```
    public void area( )
```

```
    {
```

```
        System.out.print("\n\n\t THE AREA OF THE RECTANGLE IS...."+(x*y));
```

```
    }
```

```
}
```

```
class Triangle extends Shape
{
    public void accept( )throws Exception
    {
        Scanner sc= new Scanner(System.in);

        System.out.print("\n\n\t ENTER THE BASE OF THE TRIANGLE....");

        x=sc.nextInt( );

        System.out.print("\n\n\t ENTER THE HEIGHT OF THE TRIANGLE...");

        y=sc.nextInt( );

    }

    public void area( )
    {
        System.out.print("\n\n\t THE  AREA OF THE TRIANGLE IS..." + (0.5*x*y));

    }

}

class Circle extends Shape
{
    public void accept( )throws Exception
    {
        Scanner sc= new Scanner(System.in);

        System.out.print("\n\n\t ENTER THE RADIUS OF THE CIRCLE...");

        x=sc.nextInt( );

    }

    public void area( )
    {
        System.out.print("\n\n\t THE AREA OF CIRCLE IS...." + (3.14*x*x));

    }

}
```

```

    }
}
class Shapedemo
{
    public static void main(String args[]) throws Exception
    {
        Rectangle r = new Rectangle( );
        Triangle t = new Triangle( );
        Circle c = new Circle( );
        r.accept( );
        r.area( );
        t.accept( );
        t.area( );
        c.accept( );
        c.area( );
    }
}

```

Input/Output:

ENTER THE LENGTH OF THE RECTANGLE...3

ENTER THE BREADTH OF THE RECTANGLE....4

THE AREA OF THE RECTANGLE IS....12

ENTER THE BASE OF THE TRIANGLE....4

ENTER THE HEIGHT OF THE TRIANGLE...5

THE AREA OF THE TRIANGLE IS...10.0

ENTER THE RADIUS OF THE CIRCLE...3

THE AREA OF CIRCLE IS....28.259999999999998

8. Illustrating Dynamic Binding

Aim: To develop a java program for illustrating Dynamic Binding

Program:

```
//JAVA PROGRAM TO ILLUSTRATE DYNAMIC BINDING
```

```
import java.util.*;
```

```
interface shape
```

```
{
```

```
    public void accept( );
```

```
    public void area( );
```

```
}
```

```
class Square implements shape
```

```
{
```

```
    int s;
```

```
    public void accept( )
```

```
    {
```

```
        Scanner sc=new Scanner(System.in);
```

```
        System.out.print("\n\n\t ENTER THE SIDE OF THE SQUARE...");
```

```
        s=sc.nextInt( );
```

```
    }
```

```
    public void area( )
```

```
    {
```

```
        System.out.print("\n\n\t THE AREA OF THE SQUARE IS..." + (s*s));
```

```
    }
```

```
}
```

```
class Rectangle implements shape
```

```
{
```

```
    int l,b;
```



```
public void accept( )
{
    Scanner sc= new Scanner(System.in);

    System.out.print("\n\n\t ENTER THE LENEGTH OF THE RECTANGLE....");

    l=sc.nextInt( );

    System.out.print("\n\n\t ENTER THE BREADTH OFTHE RECTANGLE....");

    b=sc.nextInt( );

}

public void area( )
{
    System.out.print("\n\n\t THE AREA OF THE RECTANGLE IS...." +(l*b));

}

}

class Demo1
{
    public static void main(String args[])throws Exception
    {
        shape sh;

        Square x=new Square( );

        Rectangle y=new Rectangle( );

        sh=x;

        sh.accept( );

        sh.area( );

        sh=y;

        sh.accept( );

        sh.area( );

    }
}
```

}

Input/Output

ENTER THE SIDE OF THE SQUARE...3

THE AREA OF THE SQUARE IS...9

ENTER THE LENEGTH OF THE RECTANGLE....4

ENTER THE BREADTH OFTHE RECTANGLE....5

THE AREA OF THE RECTANGLE IS....20

9. Illustrating Method Overloading

Aim : To develop a java program for illustrating Method Overloading

Program:

//JAVA PROGRAM TO IMPLEMENT METHOD OVERLOADING

```
import java.util.*;
```

```
class Overload
```

```
{
```

```
    public static void main(String args[])throws Exception
```

```
    {
```

```
        int s,x,ch;
```

```
        float ln,br,y;
```

```
        double a,b,c,z;
```

```
        Scanner sc=new Scanner(System.in);
```

```
    do
```

```
    {
```

```
        System.out.print("\n\n\t-----MENU-----");
```

```
        System.out.print("\n\n\t 1...AREA OF SQUARE");
```

```
        System.out.print("\n\n\t 2...AREA OF RECTANGLE");
```

```
        System.out.print("\n\n\t 3...AREA OF TRIANGLE");
```

```
        System.out.print("\n\n\t 4...EXIT");
```

```
        System.out.print("\n\n\t ENTER YOUR CHOICE...");
```

```
        ch=sc.nextInt();
```

```
        switch(ch)
```

```
        {
```

```
            case 1:
```

```
                System.out.print("\n\n\t ENTER THE SIDE OF THE SQUARE...");
```

```
                s=sc.nextInt();
```

```

        x=area(s);

        System.out.print("\n\n\t THE AREA OF SQUARE IS..." +x);

        break;

    case 2:

        System.out.print("\n\n\t ENTER THE LENGTH OF THE RECTANGLE...");

        ln=sc.nextFloat();

        System.out.print("\n\n\t ENTER THE BREADTH OF THE RECTANGLE...");

        br=sc.nextFloat();

        y=area(ln,br);

        System.out.print("\n\n\t THE AREA OF RECTANGLE IS..." +y);

        break;

    case 3:

        System.out.print("\n\n\t ENTER THE LENGTH OF SIDE-1...");

        a=sc.nextDouble();

        System.out.print("\n\n\t ENTER THE LENGTH OF SIDE-2...");

        b=sc.nextDouble();

        System.out.print("\n\n\t ENTER THE LENGTH OF SIDE-3...");

        c=sc.nextDouble();

        z=area(a,b,c);

        System.out.print("\n\n\t THE AREA OF TRIANGLE IS..." +z);

        break;

    case 4:

        System.exit(0);

    default:

        System.out.print("\n\n\t INVALID CHOICE");

}

}while(ch>=1 && ch<=3);

```

```

    }

    public static int area(int s)

    {

        int p;

        p=s*s;

        return(p);

    }

    public static float area(float ln,float br)

    {

        float p;

        p=ln*br;

        return(p);

    }

    public static double area(double a, double b, double c)

    {

        double s,p;

        s=(a+b+c)/2;

        p=Math.sqrt(s*(s-a)*(s-b)*(s-c));

        return(p);

    }

}

```

Input/Output:

-----MENU-----

1...AREA OF SQUARE

2...AREA OF RECTANGLE

3...AREA OF TRIANGLE

4...EXIT

ENTER YOUR CHOICE...1

ENTER THE SIDE OF THE SQUARE...3

THE AREA OF SQUARE IS...9

-----MENU-----

1...AREA OF SQUARE

2...AREA OF RECTANGLE

3...AREA OF TRIANGLE

4...EXIT

ENTER YOUR CHOICE...2

ENTER THE LENGTH OF THE RECTANGLE...4

ENTER THE BREADTH OF THE RECTANGLE...5

THE AREA OF RECTANGLE IS...20.0

-----MENU-----

1...AREA OF SQUARE

2...AREA OF RECTANGLE

3...AREA OF TRIANGLE

4...EXIT

ENTER YOUR CHOICE...3

ENTER THE LENGTH OF SIDE-1...4

ENTER THE LENGTH OF SIDE-2...5

ENTER THE LENGTH OF SIDE-3...6

THE AREA OF TRIANGLE IS...9.921567416492215

-----MENU-----

1...AREA OF SQUARE

2...AREA OF RECTANGLE

3...AREA OF TRIANGLE

4...EXIT

ENTER YOUR CHOICE...4

10.Illustrating Method Overriding

Aim: To develop a java program for illustrating Method Overriding

Program:

//JAVA PROGRAM TO IMPLEMENT METHOD OVERRIDING

```
import java.util.*;
```

```
class Person
```

```
{
```

```
    public void display()
```

```
    {
```

```
        System.out.print("\n\n\t PERSON");
```

```
    }
```

```
}
```

```
class Doctor extends Person
```

```
{
```

```
    public void display()
```

```
    {
```

```
        System.out.print("\n\n\t DOCTOR");
```

```
    }
```

```
}
```

```
class Override
```

```
{
```

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

```
    {
```

```
        Person p=new Person();
```

```
        Doctor d=new Doctor();
```

```
        p.display();
```

```
        d.display();
```


}

}

Input/Output:

PERSON

DOCTOR

11.Currency Conversion using Interfaces

Aim: To develop a java program for currency conversion using interfaces

Program:

// JAVA PROGRAM TO ILLUSTRATE CURRENCY CONVERSION USING
INTERFACES

```
import java.util.*;
```

```
interface convert1
```

```
{
```

```
    public void dollar_to_rupee(int d);
```

```
}
```

```
interface convert2
```

```
{
```

```
    public void euro_to_rupee(int e);
```

```
}
```

```
interface convert3
```

```
{
```

```
    public void yen_to_rupee(int y);
```

```
}
```

```
class Currency implements convert1,convert2,convert3
```

```
{
```

```
    public void dollar_to_rupee(int d )
```

```
{
```

```
        double r=74*d;
```

```
        System.out.print("\n\n\t" + d + "dollars="+r+" rupees");
```

```
}
```

```

public void euro_to_rupee(int e )
{
    double r=84.94 *e;

    System.out.print("\n\n\t" + e+"euros="+r+" rupees");
}

public void yen_to_rupee(int y)
{
    double r=0.65*y;

    System.out.print("\n\n\t" + y+"yen="+r+" rupees");
}
}

class Demo
{
    public static void main(String args[ ]) throws Exception
    {
        int p,ch;

        Scanner sc= new Scanner(System.in);

        Currency c= new Currency( );

        do
        {
            System.out.print("\n\n\t-----MENU-----");

            System.out.print("\n\n\t 1.....DOLLAR-TO-RUPEES");

            System.out.print("\n\n\t 2.....EURO-TO-RUPEES");

            System.out.print("\n\n\t 3.....YEN-TO-RUPEES");

            System.out.print("\n\n\t 4.....EXIT");

            System.out.print("\n\n\t ENTER YOUR CHOICE....");

            ch=sc.nextInt( );

```

```
switch(ch)
{
    case 1:
        System.out.print("\n\n\t ENTER THE NUMBER OF DOLLARS....");
        p=sc.nextInt( );
        c.dollar_to_rupee(p);
        break;
    case 2:
        System.out.print("\n\n\t ENTER THE NUMBER OF EUROS.....");
        p=sc.nextInt( );
        c.euro_to_rupee(p);
        break;
    case 3:
        System.out.print("\n\n\t ENTER THE NUMBER OF YEN.....");
        p=sc.nextInt( );
        c.yen_to_rupee(p);
        break;
    case 4:
        System.exit(0);
    default:
        System.out.print("\n\n\t INVALID CHOICE....");
}
}while(ch>=1 && ch<=3);
}
}
```

Input/Output:

-----MENU-----

1.....DOLLAR-TO-RUPEES

2.....EURO-TO-RUPEES

3.....YEN-TO-RUPEES

4.....EXIT

ENTER YOUR CHOICE....1

ENTER THE NUMBER OF DOLLARS....2

2dollars=148.0 rupees

-----MENU-----

1.....DOLLAR-TO-RUPEES

2.....EURO-TO-RUPEES

3.....YEN-TO-RUPEES

4.....EXIT

ENTER YOUR CHOICE....2

ENTER THE NUMBER OF EUROS.....3

3euros=254.82 rupees

-----MENU-----

1.....DOLLAR-TO-RUPEES

2.....EURO-TO-RUPEES

3.....YEN-TO-RUPEES

4.....EXIT

ENTER YOUR CHOICE....3

ENTER THE NUMBER OF YEN.....4

4yen=2.6 rupees

-----MENU-----

1.....DOLLAR-TO-RUPEES

2.....EURO-TO-RUPEES

3.....YEN-TO-RUPEES

4.....EXIT

ENTER YOUR CHOICE....4

12.Displaying Unique values from the given set of values

Aim: To develop a java program for accepting five input values between 10 and 100 and displaying the unique values from the given set of values

Program:

//JAVA PROGRAM TO ACCEPT FIVE VALUES BETWEEN 10 AND 100

//AND DISPLAY THE UNIQUE VALUES AMONG THE GIVEN VALUES

```
import java.util.*;
```

```
class Unique
```

```
{
```

```
    public static void main(String args[ ])throws Exception
```

```
    {
```

```
        int i,j,k,s,r;
```

```
        Scanner sc=new Scanner(System.in);
```

```
        s=0;
```

```
        r=0;
```

```
        int a[ ]={0,0,0,0,0};
```

```
        System.out.print("\n\n\t ENTER FIVE UNIQUE VALUES BETWEEN 10 AND 100");
```

```
        for(i=0;i<5;i++)
```

```
        {
```

```
            System.out.print("\n\n\t ENTER THE ELEMENT....");
```

```
            k=sc.nextInt( );
```

```
            if(k>=10 && k<=100)
```

```
            {
```

```
                for(j=0;j<r;j++)
```

```
                {
```

```
                    if(a[j]==k)
```

```
                    {
```

```
        s++;
    }
}
if(s>0)
{
    System.out.print("\n\n\t DUPLICATE ELEMENT");

    s--;
    i--;
    continue;
}
else
{
    a[i]=k;
    r++;
}
}
else
{
    System.out.print("\n\n\t ENTER A VALUE BETWEEN 10 AND 100");
}
}
System.out.print("\n\n\t THE ELEMENTS WITHOUT DUPLICATES ARE...");
for(i=0;i<5;i++)
{
    System.out.print(" "+a[i]);
}
}
```


}

Input/Output:

ENTER FIVE UNIQUE VALUES BETWEEN 10 AND 100

ENTER THE ELEMENT....10

ENTER THE ELEMENT....20

ENTER THE ELEMENT....20

DUPLICATE ELEMENT

ENTER THE ELEMENT....30

ENTER THE ELEMENT....40

ENTER THE ELEMENT....50

THE ELEMENTS WITHOUT DUPLICATES ARE... 10 20 30 40 50

13(a) Illustrating the usage of StringTokenizer class

Aim: To develop a java program for illustrating StringTokenizer class

Program:

```
//JAVA PROGRAM TO ILLUSTRATE THE USAGE OF STRING TOKENIZER
```

```
import java.util.*;
```

```
class Stdemo
```

```
{
```

```
    public static void main(String args[ ])throws Exception
```

```
    {
```

```
        Scanner sc=new Scanner(System.in);
```

```
        String s;
```

```
        System.out.print("\n\n\t ENTER A STRING WITH SPACES....");
```

```
        s=sc.nextLine( );
```

```
        StringTokenizer st=new StringTokenizer(s," ");
```

```
        while(st.hasMoreTokens())
```

```
        {
```

```
            System.out.print("\n\n\t "+st.nextToken( ));
```

```
        }
```

```
    }
```

```
}
```

Input/Output:

```
ENTER A STRING WITH SPACES....WELCOME TO GIST
```

```
WELCOME
```

```
TO
```

```
GIST
```

13(b) Illustrating the usage of Local Time class

//JAVA PROGRAM TO ILLUSTRATE THE USAGE OF LOCALTIME CLASS

```
import java.util.*;
import java.time.*;

class Sdemo
{
    public static void main(String args[ ])throws Exception
    {
        String s1,s2,s;

        LocalTime h,h1,h2;

        Scanner sc=new Scanner(System.in);

        System.out.print("\n\n\t ENTER THE  FIRST TIME INTERVAL IN THE FORMAT(
        HH:MM:SS).....");

        s1=sc.next( );

        h1= LocalTime.parse(s1);

        System.out.print("\n\n\t ENTER THE SECOND TIME INTERVAL IN THE
        FORMAT(HH:MM:SS).....");

        s2=sc.next( );

        h2= LocalTime.parse( s2);

        System.out.print("\n\n\t THE FIRST TIME INTERVAL IS....."+s1);

        System.out.print("\n\n\t THE SECOND TIME INTERVAL....."+s2);

        h=LocalTime.now( );

        System.out.print("\n\n\t THE CURRENT TIME IS....."+h);

        if((h.compareTo(h1) >0) && (h.compareTo(h2)<0))
        {

            System.out.print("\n\n\t THE SYSTEM TIME LIES WITHIN YOUR GIVEN
            INTERVALS");
```

```
    }  
    else  
    {  
        System.out.print("\n\n\t THE SYSTEM TIME DOES NOT LIE WITHIN YOUR  
        GIVEN INTERVALS");  
    }  
}  
}
```

Input/Output

ENTER THE FIRST TIME INTERVAL IN THE FORMAT(HH:MM:SS).....15:00:00

ENTER THE SECOND TIME INTERVAL IN THE

FORMAT(HH:MM:SS).....16:00:00

THE FIRST TIME INTERVAL IS.....15:00:00

THE SECOND TIME INTERVAL.....16:00:00

THE CURRENT TIME IS.....15:03:36.288

THE SYSTEM TIME LIES WITHIN YOUR GIVEN INTERVALS

14. Creating a User defined Exception

Aim: To develop a java program for creating a user defined exception

Program:

//JAVA PROGRAM TO CREATE A USER DEFINED EXCEPTION

```
import java.util.*;
```

```
class Simple extends Exception
```

```
{
```

```
    Simple(String s)
```

```
{
```

```
    super(s);
```

```
}
```

```
}
```

```
class Simpledemo
```

```
{
```

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

```
{
```

```
    int n;
```

```
    Scanner sc = new Scanner(System.in);
```

```
    System.out.print("\n\n\t ENTER THE AGE OF THE PERSON...");
```

```
    n=sc.nextInt( );
```

```
    if(n<18)
```

```
{
```

```
    try
```

```
{
```

```
        throw new Simple("AGE SHOULD BE GREATER THAN 18");
```

```
}
```

```
        catch(Exception e)
        {
            System.out.print(e);
        }
    }
else
{
    System.out.print("\n\n\t ELIGIBLE FOR VOTING");
}
}
```

Input/Output:

ENTER THE AGE OF THE PERSON...15

Simple: AGE SHOULD BE GREATER THAN 18

15.Performing Division using Exceptions

Aim: To develop a java program for performing division of two numbers using exceptions

Program:

```
//JAVA PROGRAM FOR HANDLING EXCEPTIONS
```

```
import java.awt.*;
```

```
import java.applet.*;
```

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

```
import javax.swing.*;
```

```
/*
```

```
<applet code="Division" width="500" height="500">
```

```
</applet>
```

```
*/
```

```
public class Division extends Applet implements ActionListener
```

```
{
```

```
    Label l1,l2,l3;
```

```
    TextField tf1, tf2, tf3;
```

```
    Button b;
```

```
    public void init( )
```

```
{
```

```
    l1= new Label("ENTER THE FIRST NUMBER");
```

```
    l2=new Label("ENTER THE SECOND NUMBER");
```

```
    l3=new Label("RATIO OF TWO NUMBERS IS");
```

```
    tf1=new TextField( );
```

```
    tf2=new TextField( );
```

```
    tf3=new TextField( );
```

```
    b= new Button("DIVIDE");
```

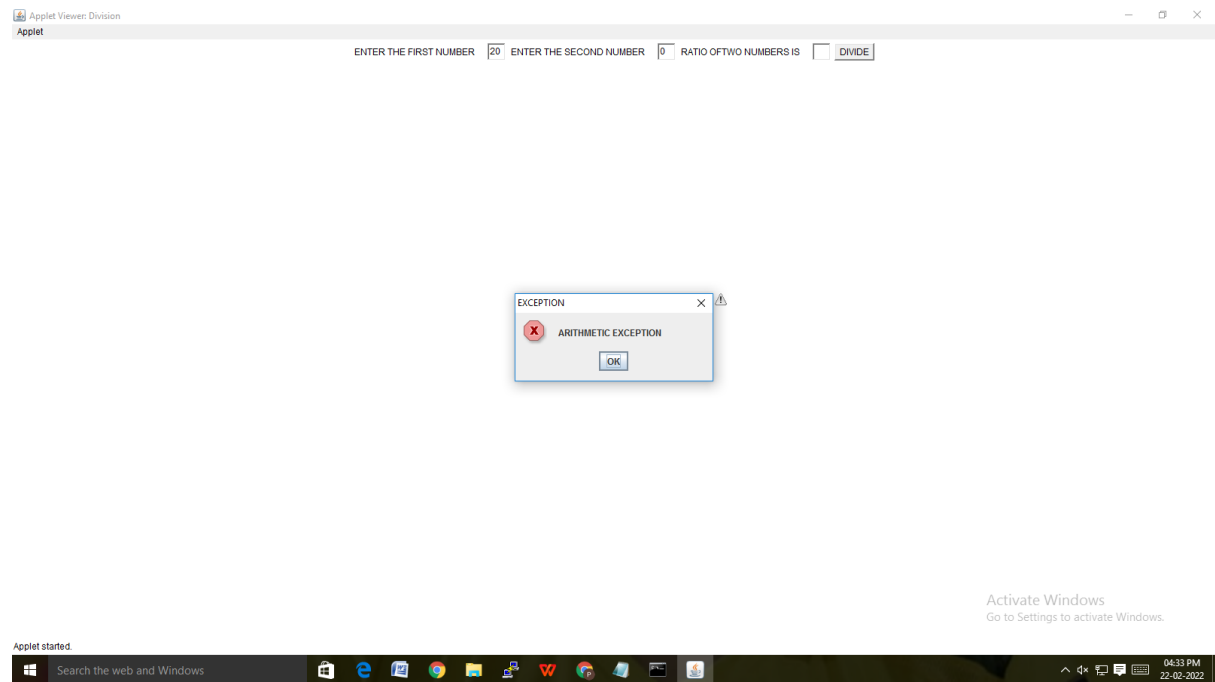
```
        add(l1);
        add(tf1);
        add(l2);
        add(tf2);
        add(l3);
        add(tf3);
        add(b);
        b.addActionListener(this);
    }
    public void actionPerformed(ActionEvent ae)
    {
        String s1=tf1.getText( );
        String s2=tf2.getText( );
        int a=Integer.parseInt(s1);
        int b=Integer.parseInt(s2);
        int c=0;

        try
        {
            c=a/b;
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(this,"ARITHMETIC
            EXCEPTION","EXCEPTION",JOptionPane.ERROR_MESSAGE);
        }
        String z=Integer.toString(c);
```



```
tf3.setText(z);  
  
}  
  
}
```

Input/Output:



16. Execution of Multiple Threads

Aim: To develop a java program for executing multiple threads without conflict

Program:

//JAVA PROGRAM FOR EXECUTING MULTIPLE THREADS WITHOUT CONFLICT

```
class Sample1 extends Thread
{
    public synchronized void run( )
    {
        while(true)
        {
            System.out.print("\n\n\t GOOD MORNING");

            try
            {
                sleep(1000);
            }
            catch(Exception e1)
            {
                System.out.print(e1);
            }
        }
    }
}

class Sample2 extends Thread
{
    public synchronized void run( )
```

```
{
    while(true)
    {
        System.out.print("\n\n\t HELLO");

        try
        {
            sleep(2000);
        }
        catch(Exception e2)
        {
            System.out.print(e2);
        }
    }
}

class Sample3 extends Thread
{
    public synchronized void run( )
    {
        while(true)
        {
            System.out.print("\n\n\t WELCOME");

            try
            {
                sleep(3000);
            }

            catch(Exception e3)
```

```

        {
            System.out.print(e3);
        }
    }
}
}
}
class Threaddemo
{
    public static void main(String args[ ])throws Exception
    {
        Sample1 s1=new Sample1( );
        Sample2 s2=new Sample2( );
        Sample3 s3=new Sample3( );
        s1.start();
        s2.start();
        s3.start();
    }
}

```

Input/Output:

GOOD MORNING

HELLO

WELCOME

GOOD MORNING

HELLO

GOOD MORNING

GOOD MORNING

HELLO

WELCOME

GOOD MORNING

GOOD MORNING

HELLO

GOOD MORNING

17. Splitting a file into parts

Aim: To develop a java program for splitting a file

Program:

```
//JAVA PROGRAM FOR SPLITTING A FILE
```

```
import java.io.*;
```

```
import java.util.*;
```

```
class Split
```

```
{
```

```
public static void main(String args[])throws Exception
```

```
{
```

```
String f,s;
```

```
Scanner sc=new Scanner(System.in);
```

```
System.out.print("\n\n\t ENTER THE FILE NAME....");
```

```
f=sc.next();
```

```
BufferedReader br1 = new BufferedReader(new FileReader(f));
```

```
int lc=0;
```

```
System.out.print("\n\n\t THE CONTENTS OF THE FILE ARE...");
```

```
while((s=br1.readLine())!=null)
```

```
{
```

```
System.out.print("\n\n\t"+s);
```

```
System.out.print("\n");
```

```
lc++;
```

```
}
```

```
System.out.println("\n\n\t NUMBER OF LINES IN THE FILE IS... " + lc);
```

```
int nof;
```

```
System.out.print("\n\n\t ENTER THE NUMBER OF FILES....");
```

```
nof=sc.nextInt();

System.out.println("\n\n\tNUMBER OF FILES TO BE GENERATED IS..." +nof);

br1.close( );

BufferedReader br2= new BufferedReader(new FileReader(f));

String sl;

int p=lc/nof;

for (int j=1;j<=nof;j++)

{

    FileWriter fw = new FileWriter("F:/File"+j+".txt");

    BufferedWriter bw = new BufferedWriter(fw);

    for (int i=1;i<=p;i++)

    {

        sl = br2.readLine();

        if (sl!= null)

        {

            bw.write(sl);

            if(i!=p)

            {

                bw.newLine();

            }

        }

    }

    bw.close();

}

System.out.print("\n\n\t THE CONTENTS OF FILE-1 ARE...\n");

BufferedReader br3=new BufferedReader(new FileReader("F:/File1.txt"));

while((s=br3.readLine())!=null)
```

```

{
    System.out.print("\n\n\t"+s);
    System.out.print("\n");
}
System.out.print("\n\n\t THE CONTENTS OF FILE-2 ARE...\n");
BufferedReader br4=new BufferedReader(new FileReader("F:/File2.txt"));
while((s=br4.readLine())!=null)
{
    System.out.print("\n\n\t"+s);
    System.out.print("\n");
}
}
}

```

Input/Output:

```

ENTER THE FILE NAME....proverb.txt
THE CONTENTS OF THE FILE ARE...
TIME IS PRECIOUS
BE BRAVE IN DIFFICULT TIMES
NUMBER OF LINES IN THE FILE IS... 2
ENTER THE NUMBER OF FILES....2
NUMBER OF FILES TO BE GENERATED IS...2
THE CONTENTS OF FILE-1 ARE...
TIME IS PRECIOUS
THE CONTENTS OF FILE-2 ARE...
BE BRAVE IN DIFFICULT TIMES

```


18. Displaying the information about the given File

Aim: To develop a java program for displaying the information about the given file

Program:

```
//JAVA PROGRAM TO DISPLAY THE INFORMATION ABOUT A FILE
```

```
import java.util.*;
```

```
import java.io.*;
```

```
class FileInfo
```

```
{
```

```
    public static void main(String args[ ])throws Exception
```

```
    {
```

```
        String s;
```

```
        Scanner sc= new Scanner(System.in);
```

```
        System.out.print("\n\n\t ENTER THE FILE NAME.....");
```

```
        s=sc.next( );
```

```
        File f1=new File(s);
```

```
        if(f1.exists( ))
```

```
        {
```

```
            System.out.print("\n\n\t FILE EXISTS");
```

```
        }
```

```
    else
```

```
    {
```

```
        System.out.print("\n\n\t FILE DOES NOT EXIST");
```

```
    }
```

```
        if(f1.canRead( ))
```

```
        {
```

```
            System.out.print("\n\n\t FILE IS READABLE");
```

```
        }
```

```
else
{
    System.out.print("\n\n\tFILE IS NOT READABLE");
}
if(f1.canWrite( ))
{
    System.out.print("\n\n\t FILE IS WRITEABLE");
}
else
{
    System.out.print("\n\n\t FILE IS NOT WRITABLE");
}

System.out.print("\n\n\t THE LENGTH OF THE FILE IS....."+f1.length( ));
}
}
```

Input/Output:

ENTER THE FILE NAME.....F:\Sdemo.java

FILE EXISTS

FILE IS READABLE

FILE IS WRITEABLE

THE LENGTH OF THE FILE IS.....1103

19. Counting the number of characters, words and lines in a file

Aim: To develop a java program for counting the number of characters, words and lines in a file

Program:

//JAVA PROGRAM FOR COUNTING THE NUMBER OF CHARACTERS, WORDS
AND LINES IN A GIVEN TEXT FILE

```
import java.io.*;

import java.util.*;

class Count

{

    public static void main(String args[ ])throws Exception

    {

        String s,cl;

        BufferedReader br=null;

        Scanner sc=new Scanner(System.in);

        System.out.print("\n\n\t ENTER THE FILE NAME...");

        s=sc.next( );

        br=new BufferedReader(new FileReader(s));

        int cc,wc,lc;

        cc=wc=lc=0;

        cl=br.readLine( );

        while(cl!=null)

        {

            lc=lc+1;

            String w[ ]=cl.split(" ");

            wc=wc+w.length;

            for(String i :w)

            {
```

```

        cc=cc+i.length( );
    }
    cl=br.readLine( );
}

System.out.print("\n\n\t THE NUMBER OFCHARACTERS IN THE FILE IS...."+cc);
System.out.print("\n\n\t THE NUMBER OF WORDS IN THE FILE IS....."+wc);
System.out.print("\n\n\t THE NUMBER OF LINES IN THE FILE IS....."+lc);
}
}

```

Input/Output:

Create a text file with the following content and name it as Proverb.txt

BE BRAVE ANDFACE THE DIFFICULT SITUATIONS

TIME IS PRECIOUS

ENTER THE FILE NAME...F:\Proverb.txt

THE NUMBER OFCHARACTERS IN THE FILE IS....50

THE NUMBER OF WORDS IN THE FILE IS.....10

THE NUMBER OF LINES IN THE FILE IS.....2

20. Displaying the contents of a file along with line number

Aim: To develop a java program for displaying the contents of a file along with line number

Program:

```
//JAVA PROGRAM TO DISPLAY THE CONTENTS OF A FILE ALONG WITH LINE  
NUMBER
```

```
import java.util.*;
```

```
import java.io.*;
```

```
class LineRead
```

```
{
```

```
    public static void main(String args[ ])throws Exception
```

```
    {
```

```
        String s,l;
```

```
        Scanner sc= new Scanner(System.in);
```

```
        System.out.print("\n\n\t ENTER THE FILE NAME....");
```

```
        s=sc.next( );
```

```
        LineNumberReader lnr = new LineNumberReader(new FileReader(s));
```

```
        while((l=lnr.readLine())!=null)
```

```
        {
```

```
            System.out.print("\n\n\t LINE-" + lnr.getLineNumber( ) + ":"+l);
```

```
        }
```

```
    }
```

```
}
```

Input/Output:

Create a text file with the following content and name it as Proverb.txt

BE BRAVE ANDFACE THE DIFFICULT SITUATIONS

TIME IS PRECIOUS

ENTER THE FILE NAME....F:\Proverb.txt

LINE-1:BE BRAVE AND FACE THE DIFFICULT SITUATIONS

LINE-2:TIME IS PRECIOUS

21. Implementing Producer-Consumer Problem

Aim: To develop a java program for implementing Producer-Consumer problem

Program:

// JAVA PROGRAM TO IMPLEMENT PRODUCER-CONSUMER PROBLEM

class Buffer

{

int n;

boolean p=false;

public synchronized int get()

{

if(p!=true)

{

try

{

wait();

}

catch(Exception e)

{

System.out.print(e);

}

}

System.out.print("\n\n\t CONSUME ITEM-"+n);

try

{

Thread.sleep(1000);

}

catch(Exception e)

```
{
    System.out.print(e);
}
p=false;
notify( );
return n;
}
public synchronized int put(int n)
{
    if(p==true)
    {
        try
        {
            wait( );
        }
        catch(Exception e)
        {
            System.out.print(e);
        }
    }
    this.n=n;
    p=true;
    System.out.print("\n\n\t PRODUCE ITEM-"+n);
    try
    {
        Thread.sleep(1000);
    }
```



```
        catch(Exception e)
        {
            System.out.print(e);
        }
        notify( );
        return n;
    }
}

class Producer implements Runnable
{
    Buffer t;
    Producer(Buffer t)
    {
        this.t=t;
        new Thread(this,"PRODUCER").start( );
    }
    public void run( )
    {
        int x=0;
        int i=0;
        while(x<10)
        {
            t.put(i++);
            x++;
        }
    }
}
```

class Consumer implements Runnable

```
{  
    Buffer t;  
    Consumer(Buffer t)  
    {  
        this.t=t;  
        new Thread(this,"CONSUMER").start( );  
    }  
    public void run( )  
    {  
        int x=0;  
        while(x<10)  
        {  
            t.get( );  
            x++;  
        }  
    }  
}
```

public class PCproblem

```
{  
    public static void main(String args[ ])  
    {  
        Buffer t=new Buffer( );  
        Producer p= new Producer(t);  
        Consumer c=new Consumer(t);  
    }  
}
```

Input/Output

PRODUCE ITEM-0

CONSUME ITEM-0

PRODUCE ITEM-1

CONSUME ITEM-1

PRODUCE ITEM-2

CONSUME ITEM-2

PRODUCE ITEM-3

CONSUME ITEM-3

PRODUCE ITEM-4

CONSUME ITEM-4

PRODUCE ITEM-5

CONSUME ITEM-5

PRODUCE ITEM-6

CONSUME ITEM-6

PRODUCE ITEM-7

CONSUME ITEM-7

PRODUCE ITEM-8

CONSUME ITEM-8

PRODUCE ITEM-9

CONSUME ITEM-9

22. Implementing a Stack using an Array

Aim: To develop a java program for implementing a stack using an array

Program:

// JAVA PROGRAM FOR IMPLEMENTING A STACK USING AN ARRAY

```
import java.util.*;

class Stack
{
    static int a[ ],top=-1,size;

    public static void main(String args[ ])throws Exception
    {
        int ch,k,x;

        Scanner sc= new Scanner(System.in);

        System.out.print("\n\n\t ENTER THE SIZE OF THE STACK.....");

        size=sc.nextInt( );

        a=new int[size];

        do
        {
            System.out.print("\n\n\t-----MENU-----");

            System.out.print("\n\n\t 1-----PUSH");

            System.out.print("\n\n\t 2-----POP");

            System.out.print("\n\n\t 3-----TRAVERSAL");

            System.out.print("\n\n\t 4-----EXIT");

            System.out.print("\n\n\t-----");

            System.out.print("\n\n\t ENTER YOUR CHOICE.....");

            ch=sc.nextInt( );

            switch(ch)
```

```
{  
    case 1:  
        System.out.print("\n\n\t ENTER THE ELEMENT TO BE PUSHED....");  
        k=sc.nextInt( );  
        x=push(k);  
        if(x!=-1)  
        {  
            System.out.print("\n\n\t" + x+ " HAS BEEN PUSHED");  
        }  
        else  
        {  
            System.out.print("\n\n\t STACK IS FULL");  
        }  
  
        break;  
    case 2:  
        x=pop( );  
        if(x!=-1)  
        {  
            System.out.print("\n\n\t"+x+"HAS BEEN POPPED");  
        }  
        else  
        {  
            System.out.print("\n\n\t STACK IS EMPTY");  
        }  
  
        break;
```

```
        case 3:
            traversal( );
            break;
        case 4:
            System.exit(0);
        default:
            System.out.print("\n\n\t INVALID CHOICE");
            break;
    }
}while(ch>=1&&ch<=3);
}
```

```
public static int push(int k)
```

```
{
    if(top==size-1)
    {
        return(-1);
    }
```

```
else
```

```
{
    top=top+1;
    a[top]=k;
    return(a[top]);
}
```

```
}
```

```
public static int pop( )
```

```
{
```

```
    if(top==-1)
    {
        return(-1);
    }
else
{
    int k=a[top];
    top=top-1;
    return(k);
}
}

public static void traversal( )
{
    if(top==-1)
    {
        System.out.print("\n\n\t THERE ARE NO ELEMENTS IN THE STACK");
    }
else
{
    System.out.print("\n\n\t THE ELEMENTS OF THE STACK ARE....");
    for(int i=top;i>=0;i--)
    {
        System.out.print(" "+a[i]);
    }
}
}
}
```

Input/Output:

ENTER THE SIZE OF THE STACK.....3

-----MENU-----

1-----PUSH

2-----POP

3-----TRAVERSAL

4-----EXIT

ENTER YOUR CHOICE.....1

ENTER THE ELEMENT TO BE PUSHED....10

10 HAS BEEN PUSHED

-----MENU-----

1-----PUSH

2-----POP

3-----TRAVERSAL

4-----EXIT

ENTER YOUR CHOICE.....3

THE ELEMENTS OF THE STACK ARE.... 10

-----MENU-----

1-----PUSH

2-----POP

3-----TRAVERSAL

4-----EXIT

ENTER YOUR CHOICE.....4

23. Performing Arithmetic Operations using User Interface Components

Aim: To develop a java program for performing Division of two numbers using User Interface Components

Program:

```
//JAVA PROGRAM FOR PERFORMING ARITHMETIC OPERATIONS USING USER  
INTERFACE COMPONENTS
```

```
import java.awt.*;
```

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

```
import java.applet.*;
```

```
/*
```

```
<applet code="Arithmetic" width="500" height="500">
```

```
</applet>
```

```
*/
```

```
public class Arithmetic extends Applet implements ActionListener
```

```
{
```

```
Label l1=new Label("ENTER FIRST NUMBER");
```

```
Label l2=new Label("ENTER SECOND NUMBER");
```

```
Label l3=new Label("RESULT IS");
```

```
TextField tf1=new TextField( );
```

```
TextField tf2=new TextField( );
```

```
TextField tf3=new TextField( );
```

```
Button b1=new Button("ADD");
```

```
Button b2=new Button("SUBTRACT");
```

```
Button b3=new Button("MULTIPLY");
```

```
Button b4=new Button("DIVIDE");
```

```
public void init( )
```

```
{
```

```
    add(l1);
```

```
    add(tf1);
```

```
    add(l2);
```

```
    add(tf2);
```

```
    add(l3);
```

```
    add(tf3);
```

```
    add(b1);
```

```
    add(b2);
```

```
    add(b3);
```

```
    add(b4);
```

```
    b1.addActionListener(this);
```

```
    b2.addActionListener(this);
```

```
    b3.addActionListener(this);
```

```
    b4.addActionListener(this);
```

```
}
```

```
public void actionPerformed(ActionEvent ae)
```

```
{
```

```
    int a,b,c;
```

```
    String s1,s2,s3;
```

```
    if(ae.getSource( )==b1)
```

```
    {
```

```
        s1=tf1.getText( );
```

```
        a=Integer.parseInt(s1);
```

```
s2=tf2.getText( );
b=Integer.parseInt( s2);
c=a+b;
s3=Integer.toString(c);
tf3.setText(s3);
}
if(ae.getSource( )==b2)
{
s1=tf1.getText( );
a=Integer.parseInt(s1);
s2=tf2.getText( );
b=Integer.parseInt( s2);
c=a-b;
s3=Integer.toString(c);
tf3.setText(s3);
}
if(ae.getSource( )==b3)
{
s1=tf1.getText( );
a=Integer.parseInt(s1);
s2=tf2.getText( );
b=Integer.parseInt( s2);
c=a*b;
s3=Integer.toString(c);
tf3.setText(s3);
}
if(ae.getSource( )==b4)
```

```
{
    s1=tf1.getText( );
    a=Integer.parseInt(s1);
    s2=tf2.getText( );
    b=Integer.parseInt( s2);
    c=a/b;
    s3=Integer.toString(c);
    tf3.setText(s3);
}
}
```

Input/Output:



Applet started.



24. Simulating Traffic Lights

Aim: To develop a java program for simulating traffic lights

Program:

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.*;

class TrafficLights implements ActionListener

{

    JFrame jf;

    JRadioButtonMenuItem r1, r2, r3;

    JLabel L1;

    TrafficLights()

    {

        jf = new JFrame();

        jf.setSize(350,200);

        jf.setTitle("Traffic Lights");

        FlowLayout FL = new FlowLayout(FlowLayout.CENTER,20,20);

        jf.setLayout(FL);

        Font f1 = new Font("Freestyle Script", Font.BOLD, 30);

        Font f2 = new Font("Freestyle Script", Font.BOLD, 20);

        Dimension d1 = new Dimension(250,50);

        Dimension d2 = new Dimension(75,30);

        L1 = new JLabel("",SwingConstants.CENTER);

        L1.setFont(f1);

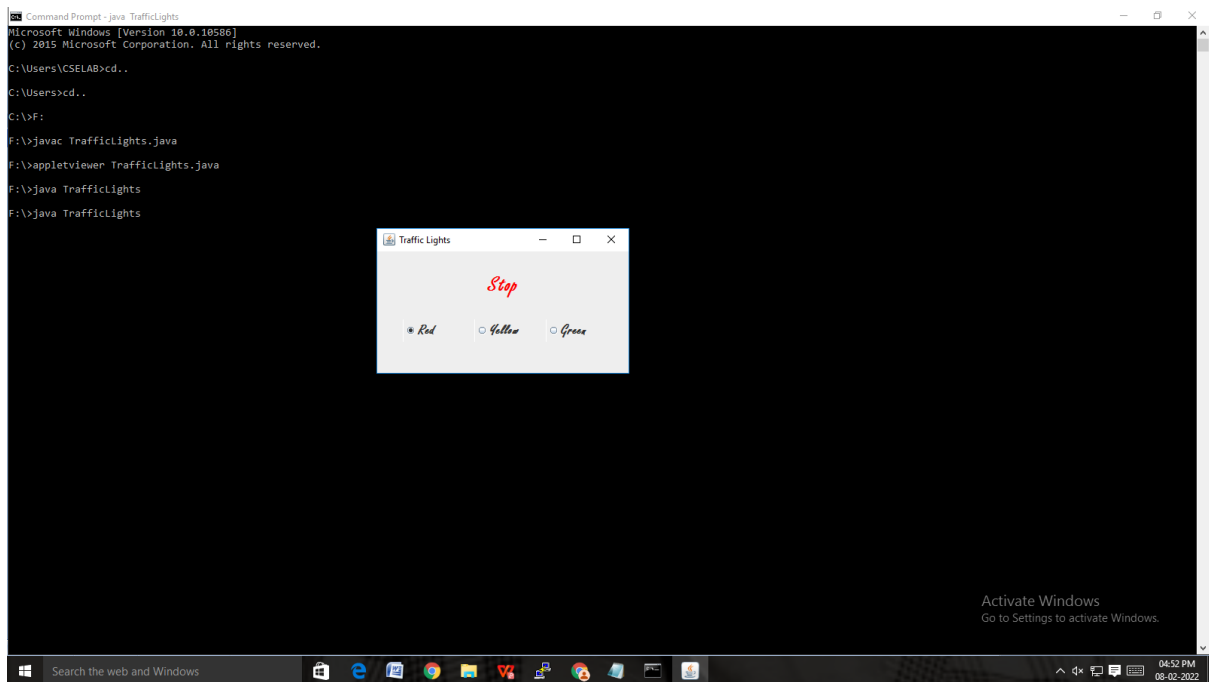
        L1.setPreferredSize(d1);

        r1 = new JRadioButtonMenuItem("Red");
```

```
r2 = new JRadioButtonMenuItem("Yellow");
r3 = new JRadioButtonMenuItem("Green");
r1.setFont(f2);
r2.setFont(f2);
r3.setFont(f2);
r1.setPreferredSize(d2);
r2.setPreferredSize(d2);
r3.setPreferredSize(d2);
ButtonGroup bg = new ButtonGroup();
bg.add(r1);
bg.add(r2);
bg.add(r3);
jf.add(L1);
jf.add(r1);
jf.add(r2);
jf.add(r3);
r1.addActionListener(this);
r2.addActionListener(this);
r3.addActionListener(this);
jf.setVisible(true);
jf.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
public void actionPerformed(ActionEvent ae)
{
    Color c1 = Color.black;
    String str = "";
```

```
        if(r1.isSelected())
        {
            str = "Stop";
            c1 = Color.red;
        }
        if(r2.isSelected())
        {
            str = "Ready";
            c1 = Color.yellow;
        }
        if(r3.isSelected())
        {
            str = "Go";
            c1 = Color.green;
        }
        L1.setText(str);
        L1.setForeground(c1);
    }
    public static void main(String as[])
    {
        new TrafficLights();
    }
}
```

Input/Output:



The screenshot displays a Windows 10 desktop environment. In the foreground, a small application window titled "Traffic Lights" is open. It features a light gray background with the word "Stop" written in a red, cursive font at the top center. Below this, there are three radio button options: "Red" (which is selected and marked with a red asterisk), "Yellow", and "Green". In the background, a Command Prompt window is visible, showing the following commands and their outputs:

```
Microsoft Windows [Version 10.0.18586]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\CSELAB>cd..
C:\Users>cd..
C:\>F:
F:\>javac TrafficLights.java
F:\>appletviewer TrafficLights.java
F:\>java TrafficLights
F:\>java TrafficLights
```

The Windows taskbar at the bottom shows the Start button, a search bar, and several pinned application icons including File Explorer, Edge, and various utility programs. The system tray on the right indicates the time as 04:52 PM on 08-02-2022. An "Activate Windows" watermark is visible in the bottom right corner of the desktop area.

25. Waving a Flag

Aim: To develop a java program for waving a flag

Program:

//JAVA PROGRAM FOR WAVING A FLAG USING APPLETS AND THREADS

```
import java.awt.*;
```

```
import java.applet.*;
```

```
import java.awt.geom.*;
```

```
import java.awt.image.*;
```

```
/*
```

```
<applet code="Flag" width="500" height="500">
```

```
</applet>
```

```
*/
```

```
public class Flag extends Applet implements Runnable
```

```
{
```

```
    int ah,aw;
```

```
    Graphics gimg;
```

```
    Image img;
```

```
    Thread t;
```

```
    int x,y;
```

```
    int xinc,yinc;
```

```
    public void init( )
```

```
    {
```

```
        aw=getSize( ).width;
```

```
        ah=getSize( ).height;
```

```
        xinc=10;
```

```
        yinc=10;
```

```
        x=101;
```

```
        y=151;

        img=createImage(aw,ah);

        gimg=img.getGraphics( );

        t= new Thread(this);

    }

    public void start( )

    {

        t.start( );

    }

    public void run( )

    {

        while(true)

        {

            x=x+xinc;

            y=y+yinc;

            if(x<=60 || (x+xinc)>=130)

            {

                xinc=xinc*(-1);

            }

            if(y<=50 || (y+yinc)>=230)

            {

                yinc=yinc*(-1);

            }

            repaint( );

            try

            {

                Thread.sleep(1000);
```

```

    }

    catch(Exception e)

    {

        System.out.print(e);

    }

}

public void paint(Graphics g)

{

    Graphics2D gimg=(Graphics2D) g;

    Color c1=Color.RED;

    Color c2=Color.ORANGE;

    Rectangle2D rect=new Rectangle2D.Double(50,50,10,300);

    CubicCurve2D cub1= new CubicCurve2D.Double(60,50,x,10,y,100,200,50);

    gimg.draw(cub1);

    CubicCurve2D cub2=new CubicCurve2D.Double(60,100,x,60,y,150,200,100);

    gimg.draw(cub2);

    CubicCurve2D cub3=new CubicCurve2D.Double(60,150,x,110,y,200,200,150);

    gimg.draw(cub3);

    CubicCurve2D cub4=new CubicCurve2D.Double(60,200,x,160,y,250,200,200);

    gimg.draw(cub3);

    Line2D l1=new Line2D.Double(200,50,200,200);

    gimg.draw(l1);

    GeneralPath gp1=new GeneralPath( );

    GeneralPath gp2=new GeneralPath( );

    gp1.moveTo(60,50);

    gp1.curveTo(x,10,y,100,200,50);

```

```

gp1.lineTo(200,100);
gp1.curveTo(y,150,x,60,60,100);
gp1.lineTo(60,50);
gimg.setColor(new Color(255,153,51));
gimg.fill(gp1);
gimg.setColor(Color.BLACK);
gimg.fill(rect);
gp2.moveTo(60,150);
gp2.curveTo(x,110,y,200,200,150);
gp2.lineTo(200,200);
gp2.curveTo(y,250,x,160,60,200);
gp2.lineTo(60,150);
gimg.setColor(Color.GREEN);
gimg.fill(gp2);
}
}

```

Input/Output:



Applet started

26. Converting Decimal Number to Binary Number using User Interface Components

Aim: To develop a java program for converting a decimal number into binary number using user interface components.

Program:

```
//JAVA PROGRAM TO CONVERT DECIMAL NUMBER INTO BINARY NUMBER
```

```
import java.awt.*;
```

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

```
import java.applet.*;
```

```
/*
```

```
<applet code="Convert" width="500" height="500">
```

```
</applet>
```

```
*/
```

```
public class Convert extends Applet implements ActionListener
```

```
{
```

```
    Label l1= new Label("ENTER A DECIMAL NUMBER");
```

```
    Label l2= new Label("THE EQUIVALENT BINARY NUMBER IS");
```

```
    TextField tf1=new TextField(10);
```

```
    TextField tf2=new TextField(10);
```

```
    Button b=new Button("CONVERT");
```

```
    public void init( )
```

```
    {
```

```
        add(l1);
```

```
        add(tf1);
```

```
        add(l2);
```

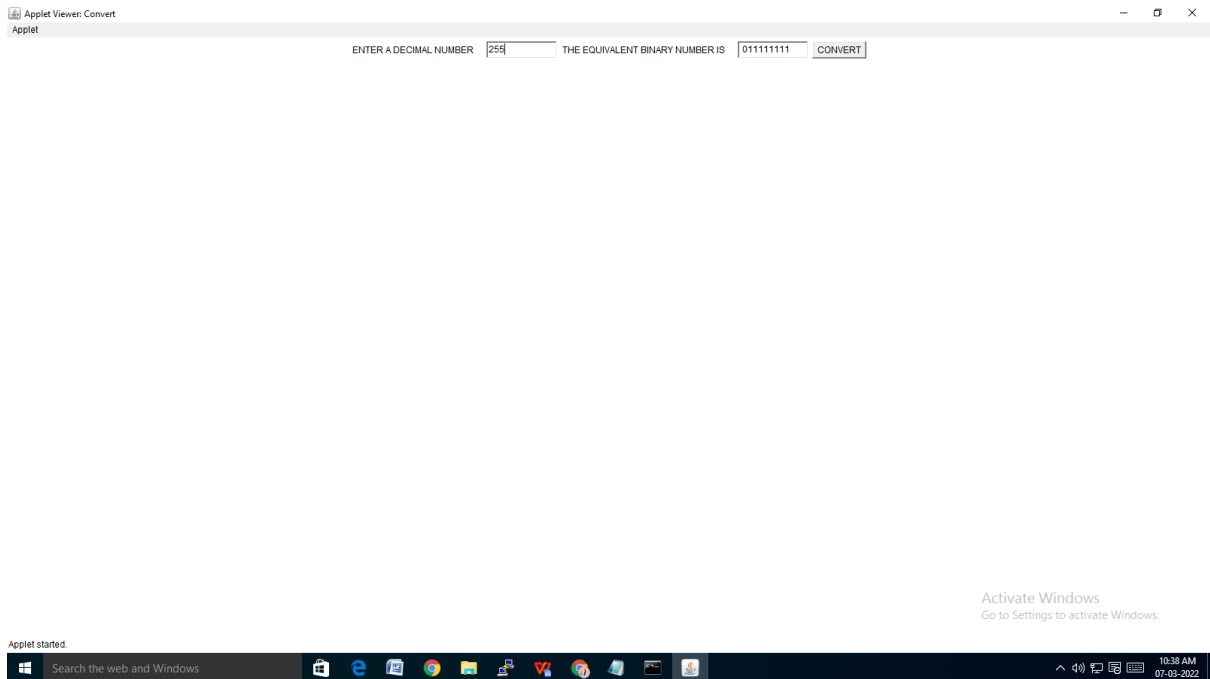
```
        add(tf2);
```

```
        add(b);
```

```
        b.addActionListener(this);
```

```
}  
  
public void actionPerformed(ActionEvent ae)  
{  
    String s,t;  
    int i,j,p,r;  
    int a[ ]=new int[20];  
    t=" ";  
    s=tf1.getText( );  
    p=Integer.parseInt(s);  
    i=0;  
    while(p>0)  
    {  
        r=p%2;  
        a[i]=r;  
        i=i+1;  
        p=p/2;  
    }  
    for(j=i;j>=0;j--)  
    {  
        t=t+Integer.toString(a[j]);  
    }  
    tf2.setText(t);  
}  
}
```

Input/Output:



27. Handling Mouse Events

Aim: To develop a java program for handling Mouse Events

Program:

```
//JAVA PROGRAM TO HANDLE MOUSE EVENTS
```

```
import java.awt.*;
```

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

```
import java.applet.*;
```

```
/*
```

```
<applet code="Mousedemo" width="500" height="500">
```

```
</applet>
```

```
*/
```

```
public class Mousedemo extends Applet implements MouseListener, MouseMotionListener
```

```
{
```

```
    int x=250,y=250;
```

```
    String s;
```

```
    public void init( )
```

```
    {
```

```
        s=" ";
```

```
        addMouseListener(this);
```

```
        addMouseMotionListener(this);
```

```
        setBackground(Color.cyan);
```

```
        setForeground(Color.black);
```

```
    }
```

```
    public void mouseEntered(MouseEvent me)
```

```
    {
```

```
        s="MOUSE ENTERED";
```

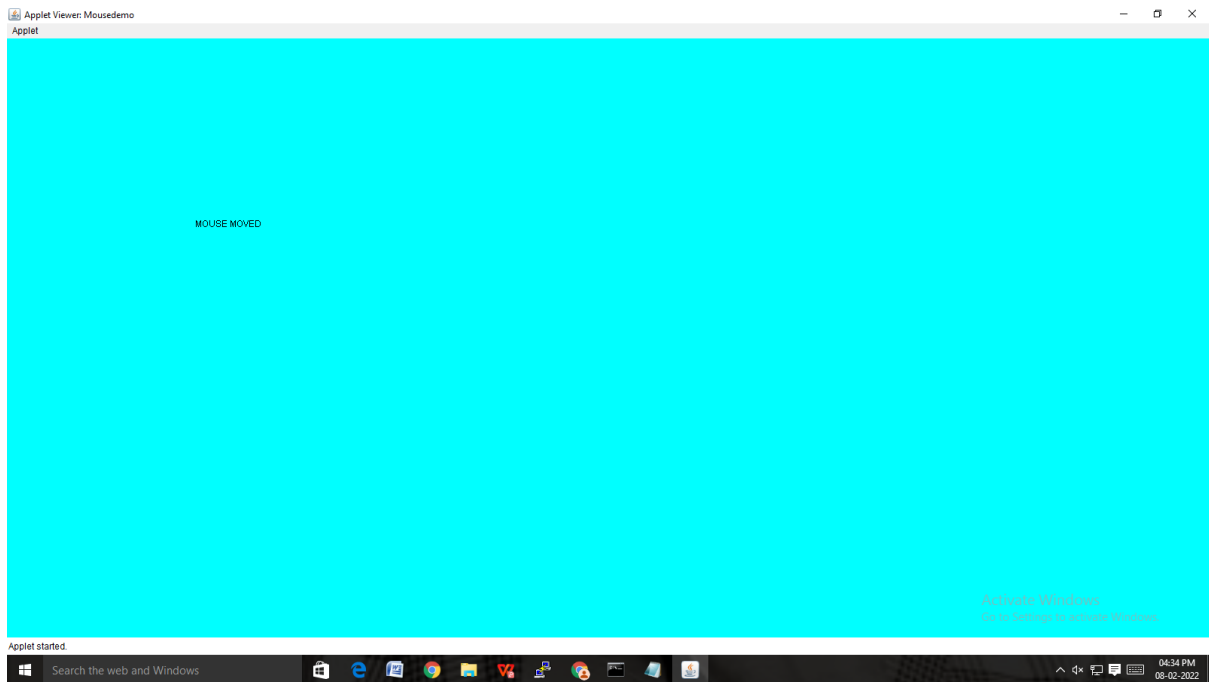
```
        repaint( );
```



```
    }  
  
    public void mouseExited(MouseEvent me)  
    {  
        s="MOUSE EXITED";  
        repaint( );  
    }  
  
    public void mousePressed(MouseEvent me)  
    {  
        s="MOUSE PRESSED";  
        repaint( );  
    }  
  
    public void mouseReleased(MouseEvent me)  
    {  
        s="MOUSE RELEASED";  
        repaint();  
    }  
  
    public void mouseMoved(MouseEvent me)  
    {  
        s="MOUSE MOVED";  
        repaint( );  
    }  
  
    public void mouseDragged(MouseEvent me)  
    {  
        s="MOUSE DRAGGED";  
        repaint( );  
    }  
  
    public void mouseClicked(MouseEvent me)
```

```
{  
    s="MOUSECLICKED";  
    repaint( );  
}  
public void paint(Graphics g)  
{  
    g.drawString(s,x,y);  
}  
}
```

Input/Output:



28. Handling Key Events

Aim: To develop a java program for handling key events

Program:

```
//JAVA PROGRAM FOR HANDLING KEY EVENTS
```

```
import java.applet.*;
```

```
import java.awt.*;
```

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

```
/*
```

```
<applet code="Keydemo" width="500" height="500">
```

```
</applet>
```

```
*/
```

```
public class Keydemo extends Applet implements KeyListener
```

```
{
```

```
    int x=100,y=100;
```

```
    String s=" ";
```

```
    public void init( )
```

```
{
```

```
    setBackground(Color.ORANGE);
```

```
    setForeground(Color.BLACK);
```

```
    addKeyListener(this);
```

```
    requestFocus( );
```

```
}
```

```
public void keyPressed(KeyEvent ke)
```

```
{
```

```
    s="KEY DOWN";
```

```
    showStatus(s);
```

```
}
```

```

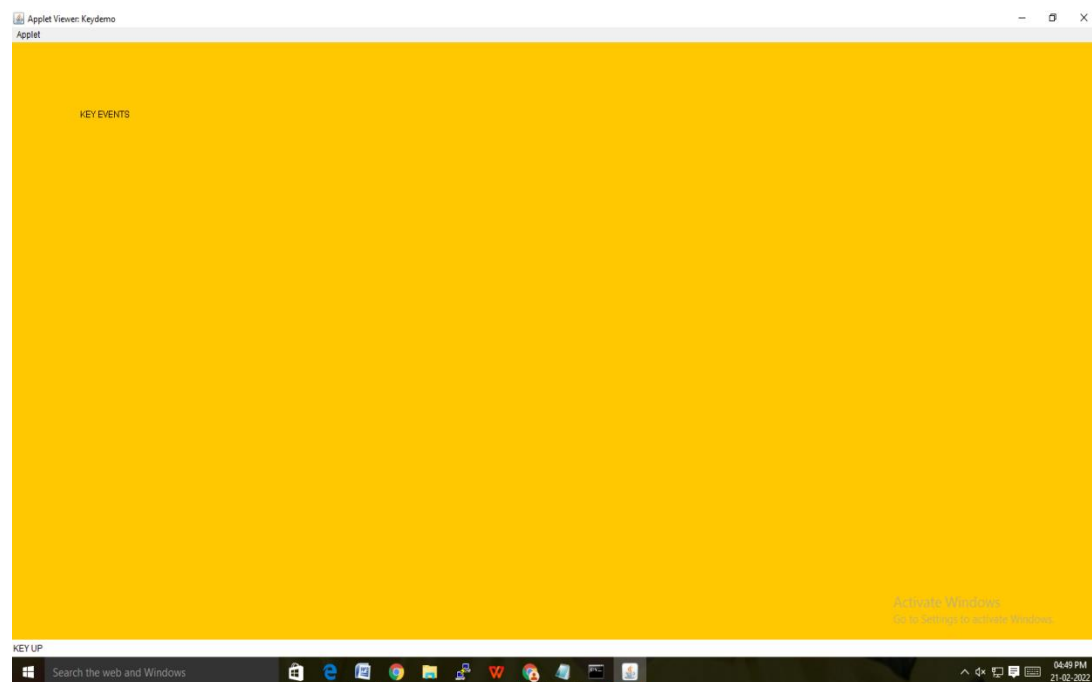
public void keyReleased(KeyEvent ke)
{
    s="KEY UP";
    showStatus(s);
}

public void keyTyped(KeyEvent ke)
{
    s=s+ke.getKeyChar( );
    showStatus(s);
}

public void paint(Graphics g)
{
    g.drawString("KEY EVENTS",x,y);
}
}

```

Input/Output:



29. Finding Maximum and Minimum using Generics

Aim: To develop a java program for finding the maximum and minimum of the given elements using Generics

Program:

//JAVA PROGRAM TO FIND THE MAXIMUM AND MINIMUM OF THE GIVEN ELEMENTS USING GENERICS

```
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 Gendemo  
{  
  
    public static void main(String args[])throws Exception  
    {  
  
        int i;  
  
        Integer a[]={ 10,2,5,4,6,1};  
  
        Character b[]={ 'v','p','s','a','n','h'};  
  
        Double c[]={ 20.2,45.4,71.6,88.3,54.6,10.4};  
  
        MyClass<Integer> iob = new MyClass<Integer>(a);  
  
        MyClass<Character> cob = new MyClass<Character>(b);  
  
        MyClass<Double>dob = new MyClass<Double>(c);  
  
        System.out.print("\n\n\t THE VALUES IN THE INTEGER ARRAY ARE....");  
  
        for(i=0;i<a.length;i++)  
        {  
  
            System.out.print(" "+a[i]);  
  
        }  
  
        System.out.print("\n\n\t THE VALUES IN THE CHARACTER ARRAY ARE....");  
  
        for(i=0;i<a.length;i++)  
        {  
  
            System.out.print(" "+b[i]);  
  
        }  
  
        System.out.print("\n\n\t THE VALUES IN THE DOUBLE ARRAY ARE....");  
  
        for(i=0;i<a.length;i++)  
        {  
  
            System.out.print(" "+b[i]);  
  
        }  
  
    }  
}
```

```
}
```

```
System.out.println("\n\n\t MAXIMUM VALUE IN INTEGER ARRAY IS... " + iob.max());
```

```
System.out.println("\n\n\t MINIMUM VALUE IN INTEGER ARRAY IS..." + iob.min());
```

```
System.out.println("\n\n\t MAXIMUM VALUE IN CHARACTER ARRAY IS..." +  
cob.max());
```

```
System.out.println("\n\n\t MINIMUM VALUE IN CHARACTER ARRAY IS..." +  
cob.min());
```

```
System.out.println("\n\n\t MAXIMUM VALUE IN DOUBLE ARRAY IS....." + dob.max());
```

```
System.out.println("\n\n\t MINIMUM VALUE IN DOUBLE ARRAY IS..... " + dob.min());
```

```
}
```

```
}
```

Input/Output:

THE VALUES IN THE INTEGER ARRAY ARE.... 10 2 5 4 6 1

THE VALUES IN THE CHARACTER ARRAY ARE.... v p s a n h

THE VALUES IN THE DOUBLE ARRAY ARE.... v p s a n h

MAXIMUM VALUE IN INTEGER ARRAY IS... 10

MINIMUM VALUE IN INTEGER ARRAY IS...1

MAXIMUM VALUE IN CHARACTER ARRAY IS...v

MINIMUM VALUE IN CHARACTER ARRAY IS....a

MAXIMUM VALUE IN DOUBLE ARRAY IS.....88.3

MINIMUM VALUE IN DOUBLE ARRAY IS..... 10.4

30. JDBC program for insert and view records in a table

Aim: To develop a java program for inserting a record into a table

Program:

```
import java.sql.*;

class OracleCon1
{
    public static void main(String args[])throws Exception
    {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con=DriverManager.getConnection(
            "jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
        Statement stmt=con.createStatement();
        int m=stmt.executeUpdate("insert into student values(1000,'SURESH',75)");
        if(m==1)
        {
            System.out.print("\n\n\t RECORD HAS BEEN INSERTED SUCCESSFULLY");
        }
        else
        {
            System.out.print("\n\n\t RECORD HAS NOT BEEN INSERTED");
        }
        con.close();
    }
}
```

Input/output:

RECORD HAS BEEN INSERTED SUCCESSFULLY

Aim: To develop a java program for viewing records in a table

Program:

// JAVA PROGRAM FOR VIEWING RECORDS IN A TABLE

```
import java.sql.*;

class OracleCon2
{
public static void main(String args[])throws Exception
{
Class.forName("oracle.jdbc.driver.OracleDriver");
Connection con=DriverManager.getConnection(
"jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from student");
while(rs.next())
{
System.out.println(rs.getInt(1)+" "+rs.getString(2)+" "+rs.getString(3));
}
con.close();
}
}
```

Input/output:

1000 SURESH 75

ADDITIONAL PROGRAMS

1. Generation of N-terms of Fibonacci series using Recursion

Aim: To develop a java program for generating N-terms of Fibonacci series using Recursion

Program:

//JAVA PROGRAM FOR GENERATING N-TERMS OF FIBONACCI SERIES USING RECURSION

```
import java.util.*;

class Fibonacci

{

    public static void main(String args[ ])throws Exception

    {

        int n;

        Scanner sc= new Scanner(System.in);

        System.out.print("\n\n\t ENTER THE NUMBER OFTERMS YOU WANT...");

        n=sc.nextInt( );

        System.out.print("\n\n\t THE "+n+"-TERMS OF FIBONACCI SERIES ARE...");

        for(int i=0;i<n;i=i+1)

        {

            System.out.print(fib(i) + " ");

        }

    }

    public static int fib(int n)

    {

        if(n==0)

        {

            return(0);

        }

        else if(n==1)
```

```
{  
    return(1);  
}  
else  
{  
    return(fib(n-2)+fib(n-1));  
}  
}  
}
```

Input/Output:

ENTER THE NUMBER OF TERMS YOU WANT...7

THE 7-TERMS OF FIBONACCI SERIES ARE...0 1 1 2 3 5 8

2. Implementing Multiple Inheritance

Aim: To develop a java program for implementing Multiple Inheritance

Program:

//JAVA PROGRAM TO IMPLEMENT MULTIPLE INHERITANCE

```
import java.util.*;

class Student
{
    int rollno;

    String name;

    public void accept_details( )throws Exception
    {
        Scanner sc=new Scanner(System.in);

        System.out.print("\n\n\t ENTER THE ROLLNO.....");

        rollno=sc.nextInt( );

        System.out.print("\n\n\t ENTER THE NAME.....");

        name=sc.next( );
    }

    public void display_details( )
    {
        System.out.print("\n\n\t ROLLNO IS...."+rollno);

        System.out.print("\n\n\t NAME IS....."+name);
    }
}

class Test extends Student
{
    int m,p,c;

    public void accept_marks( )throws Exception
```

```

{
    Scanner sc=new Scanner(System.in);

    System.out.print("\n\n\t ENTER THE MARKS IN MATHS....");

    m=sc.nextInt( );

    System.out.print("\n\n\t ENTER THE MARKS IN PHYSICS...");

    p=sc.nextInt( );

    System.out.print("\n\n\t ENTER THE MARKS IN CHEMISTRY...");

    c=sc.nextInt( );

}

public void display_marks( )

{

    System.out.print("\n\n\t MARKS IN MATHS...."+m);

    System.out.print("\n\n\t MARKS IN PHYSICS..."+p);

    System.out.print("\n\n\t MARKS IN CHEMISTRY..."+c);

}

}

interface Sports

{

    public int spwt=5;

    public void display_spwt( );

}

class Result extends Test implements Sports

{

    int total;

    public void display_spwt( )

    {

```

```
        System.out.print("\n\n\t SPORTS WEIGHTAGE...."+spwt);
    }

    public void calculate( )
    {
        total=m+p+c+spwt;
    }

    public void display_total( )
    {
        System.out.print("\n\n\t TOTAL MARKS...."+total);
    }
}

class Resultdemo
{
    public static void main(String args[ ])throws Exception
    {
        Result r=new Result( );
        r.accept_details( );
        r.accept_marks( );
        r.display_details( );
        r.display_marks( );
        r.display_spwt( );
        r.calculate( );
        r.display_total( );
    }
}
```

Input/Output:

ENTER THE ROLLNO.....1000

ENTER THE NAME.....SURESH

ENTER THE MARKS IN MATHS....75

ENTER THE MARKS IN PHYSICS...75

ENTER THE MARKS IN CHEMISTRY...75

ROLLNO IS....1000

NAME IS.....SURESH

MARKS IN MATHS....75

MARKS IN PHYSICS...75

MARKS IN CHEMISTRY...75

SPORTS WEIGHTAGE....5

TOTAL MARKS....230