PROGRAM 1

AIM

Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

ALGORITHM

Step 1: Start.

Step 2: Define a class having name Product and members as pcode, pname and price.

Step 3: Declare three objects in the class and add the values of each data members into objects.

Step 4: Using if condition check which object has the lowest price and print it.

Step 5: Stop.

PROGRAM CODE

product.java	public class product {
	int pcode;
	String pname;
	int price;
	
	public static void main(String[] args) {
	product obj1=new product();
	obj1.pcode=1001;
	obj1.pname="BMW";
	obj1.price=2500000;
	product obj2=new product();
	obj2.pcode=2012;
	obj2.pname="Benz";
	obj2.price=2000000;
	product obj3=new product();
	obj3.pcode=3211;
	obj3.pname="Hyundai";
	obj3.price=1500000;
	if(obj1.price<=obj2.price && obj1.price<=obj3.price)
	System.*out*.println(obj1.pname+" is cheaper than "+obj2.pname+" and "+obj3.pname);

```
        else if(obj2.price<=obj1.price && obj2.price<=obj3.price)
        System.out.println(obj2.pname+" is cheaper than "+obj1.pname+"
and "+obj3.pname);
        else
                System.out.println(obj3.pname+" is cheaper than
"+obj1.pname+" and "+obj2.pname);

        }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> product [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.ope
Hyundai is cheaper than BMW and Benz
```

PROGRAM 2

AIM

Read 2 matrices from the console and perform matrix addition.

ALGORITHM

Step 1: Start.

Step 2: Define a class having name AddMatrix.

Step 3: Read row number(m), column number (n) and initialize the double dimensional arrays mat1[][], mat2[][], res[][] with same row number ,column number.

Step 4: Store the first matrix elements into the two-dimensional array matrix mat1[][] using two for loops. i indicates row number, j indicates column index. Similarly second matrix elements in to mat2[][].

Step 5: Add the two matrices using for loop.
for i=0 to i<m
for j=0 to j<n
mat1[i][j] + mat2[i][j] and store it in to the matrix res[i][j] .

Step 6: Print sum of matrices res[i][j].

Stop 7: Stop.

PROGRAM CODE

matrix_add.java

```
import java.util.Scanner;
public class matrix_add {

        public static void main(String[] args) {
                int m,n,i,j;
                Scanner in = new Scanner(System.in);

                System.out.println("Enter the number of rows");
                m = in.nextInt();

                System.out.println("Enter the number columns");
                n = in.nextInt();

                int mat1[][] = new int[m][n];
                int mat2[][] = new int[m][n];
```

```
                int result[][] = new int[m][n];

                System.out.println("Enter the elements of matrix1");

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

                for ( j= 0 ; j < n ;j++ )
                mat1[i][j] = in.nextInt();

                System.out.println();
                }
                System.out.println("Enter the elements of matrix2");

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

                for ( j= 0 ; j < n ;j++ )
                mat2[i][j] = in.nextInt();

                System.out.println();
                }

                for ( i= 0 ; i < m ; i++ )
                for ( j= 0 ; j < n ;j++ )
                result[i][j] = mat1[i][j] + mat2[i][j] ;

                System.out.println("Sum of matrices:-");

                for ( i= 0 ; i < m ; i++ )
                {
                for ( j= 0 ; j < n ;j++ )
                System.out.print(result[i][j]+"\t");

                System.out.println();
                }

        }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> matrix_add [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.c
Enter the number of rows
3
Enter the number columns
3
Enter the elements of matrix1
2 4 5

6 9 5

1 3 6

Enter the elements of matrix2
6 5 7

9 6 1

8 7 0

Sum of matrices:-
8       9       12
15      15      6
9       10      6
```

PROGRAM 3

AIM

Add complex numbers.

ALGORITHM

Step 1: Start.

Step 2: Define a class having name ComplexNumber and data members are real and imaginary number.

Step 3: Define a function ComplexNumber and add values to variables.

Step 4: Define a function ComplexNumber sum to add complex number using 3rd ComplexNumber object and return the value.

Step 5: Print the sum value.

Step 6: Stop.

PROGRAM CODE

Complex.java	public class Complex {
	double real;
	double img;
	
	Complex(double r, double i){
	this.real = r;
	this.img = i;
	}
	
	public static Complex sum(Complex c1,Complex c2)
	{
	
	Complex temp = new Complex(0, 0);
	
	temp.real = c1.real + c2.real;
	temp.img = c1.img + c2.img;
	return temp;
	}
	public static void main(String[] args) {
	Complex c1 = new Complex(8.2, 6);
	Complex c2 = new Complex(6.5, 3.7);

```
                Complex temp = sum(c1, c2);
                System.out.printf("Sum is: "+ temp.real+" + "+ temp.img +"i");
                    // TODO Auto-generated method stub

            }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Complex (1) [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj
Sum is: 14.7 + 9.7i
```

PROGRAM 4

AIM

Read a matrix from the console and check whether it is symmetric or not.

ALGORITHM

Step 1: Start.

Step 2 : Read row number,column number and initialize the double dimensional array with same row number ,column number.

Step 3 : Store the first matrix elements into the two-dimensional array matrix using two for loops. i indicates row number, j indicates column index.

Step 4: Check whether the matrix is symmetric or not.

Step 5: Print the symmetric matrix or if not.

Step 6: Stop.

PROGRAM CODE

Matrix_symmetric.java	import java.util.Scanner;
	public class Matrix_symmetric {
	
	public static void main(String[] args) {
	Scanner mat = new Scanner(System.*in*);
	
	System.*out*.println("Enter the no. of rows : ");
	
	int rows = mat.nextInt();
	
	System.*out*.println("Enter the no. of columns : ");
	
	int cols = mat.nextInt();
	
	int matrix[][] = new int[rows][cols];
	
	System.*out*.println("Enter the elements :");
	
	for (int i = 0; i < rows; i++)
	{
	for (int j = 0; j < cols; j++)
	{

```
                matrix[i][j] = mat.nextInt();
            }
        }

        System.out.println("Input matrix :");

        for (int i = 0; i < rows; i++)
        {
            for (int j = 0; j < cols; j++)
            {
                System.out.print(matrix[i][j]+"\t");
            }

            System.out.println();
        }


        if(rows != cols)
        {
            System.out.println("Matrix is not a square matrix, It is not
symmetric.");
        }
        else
        {
            boolean symmetric = true;

            for (int i = 0; i < rows; i++)
            {
                for (int j = 0; j < cols; j++)
                {
                    if(matrix[i][j] != matrix[j][i])
                    {
                        symmetric = false;
                        break;
                    }
                }
            }

            if(symmetric)
            {
                System.out.println("Entered matrix is symmetric...");
            }
            else
            {
                System.out.println("The given matrix is not symmetric...");
            }
        }

        mat.close();


                    // TODO Auto-generated method stub

            }
```

	}

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Matrix_symmetric [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclips
Enter the no. of rows :
3
Enter the no. of columns :
3
Enter the elements :
2 4 6
7 5 3
9 7 5
Input matrix :
2       4       6
7       5       3
9       7       5
The given matrix is not symmetric...
```

PROGRAM 5

AIM

Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.

ALGORITHM

Step 1: Start.

Step 2: Define a class cpu with data member price and class processor.

Step 3: Class processor contain data members no_cores,manufacturer and a nested class RAM.

Step 4: class RAM contain memory and manufacturer as data members.

Step 5: Create objects in corresponding classes and display it's details.

Step 6: Stop.

PROGRAM CODE

CPU1.java

```
package CPU;
class CPU {
            double price;
   class Processor{

      double cores=3.2;
      String manufacturer="intel";

      double getCache(){
         return 4.5;
   }
   }
   protected class RAM{


      double memory=256;
      String manufacturer="intel";
      double getClockSpeed(){
```

```
            return 5.8;
        }
    }
}
public class CPU1 {
            public static void main(String[] args) {
                        CPU cpu = new CPU();

        CPU.Processor processor = cpu.new Processor();

        CPU.RAM ram = cpu.new RAM();

        System.out.println("Processor name="+processor.manufacturer);
        System.out.println("Processor core="+processor.cores);
        System.out.println("Ram name = " + ram.manufacturer);
        System.out.println("Ram memory space="+ram.memory);
        System.out.println("Processor Cache = " + processor.getCache());
        System.out.println("Ram Clock speed = " + ram.getClockSpeed());
                        // TODO Auto-generated method stub

            }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> CPU1 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.openjc
Processor name=intel
Processor core=3.2
Ram name = intel
Ram memory space=256.0
Processor Cache = 4.5
Ram Clock speed = 5.8
```

PROGRAM 6

AIM

Program to Sort strings.

ALGORITHM

Step 1: Start

Step 2: Check each element in the given list with the string provided by the user.

Step 3: If string is found, display the position of the string found, else display string not found.

Step 4: Stop

PROGRAM CODE

SortingStrings.java	

```
import java.util.Scanner;
public class SortingStrings {

        public static void main(String[] args) {
                int n;
    String temp;
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter number of strings ");
    n=sc.nextInt();


    String str[] = new String[n];
    Scanner sc1 = new Scanner(System.in);

    System.out.println("Enter the Strings");
    for(int i=0;i<n;i++)
    {
      str[i]= sc1.nextLine();
    }
    sc.close();
    sc1.close();

    for(int i=0;i<n; i++)
    {
      for (int j=i+1;j<n; j++) {
        if (str[i].compareTo(str[j])>0)
        {
```

```
                    temp = str[i];
                    str[i] = str[j];
                    str[j] = temp;
                }
            }
        }

        System.out.print("Strings in Sorted Order:\n");
        for(int i=0;i<n;i++)
        {
            System.out.print(str[i]+ "\t ");
        }

            }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> SortingStrings [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.openj
Enter number of strings 7
Enter the Strings
Ronaldo
Messi
Neymar
Mbappe
David Beckham
Pele
Mardona
Strings in Sorted Order:
David Beckham    Mardona         Mbappe  Messi   Neymar  Pele    Ronaldo
```

PROGRAM 7

AIM

Search an element in an array.

ALGORITHM

Step 1: Start

Step 2: Select the first element of the list (i.e., Element at first position in the list).

Step 3: Compare the selected element with all the other elements in the list.

Step 4: In every comparison, if any element is found smaller than the selected element (for Ascending order), then both are swapped.

Step 5: Repeat the same procedure with element in the next position in the list till the entire list is sorted.

Step 6: Stop

PROGRAM CODE

Array_search.java	import java.util.Scanner;
	
	public class array_search {
	
	public static void main(String[] args) {
	int n, x, flag = 0, i = 0;
	Scanner sc = new Scanner(System.*in*);
	System.*out*.print("Enter no. of elements ");
	n = sc.nextInt();
	int a[] = new int[n];
	System.*out*.println("Enter the element:");
	for(i = 0; i < n; i++)
	{
	a[i] = sc.nextInt();
	}
	System.*out*.print("Enter the element to search:");
	x = sc.nextInt();
	sc.close();
	for(i = 0; i < n; i++)
	{
	if(a[i] == x)
	{
	flag = 1;
	break;

```
            }
            else
            {
                flag = 0;
            }
        }
        if(flag == 1)
        {
            System.out.println("Element found at position:"+(i + 1));
        }
        else
        {
            System.out.println("Element not found");
        }
         }
         // TODO Auto-generated method stub
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> array_search [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj
Enter no. of elements 10
Enter the element:
55
74
36
85
9
17
39
70
200
59
Enter the element to search:74
Element found at position:2
```

PROGRAM 8

AIM

Perform string manipulations

ALGORITHM

Step 1: Start

Step 2: Take the strings provided by the user and concatenate them.

Step 3: Display the combined string with lower case.

Step 3: Display the combined string with upper case.

Step 4: Display the combined string after replacing all the ‘s’ & ‘S’ characters with ‘$’ character.

Step 5: Stop

PROGRAM CODE

String_Manip.java	public class String_Manip {
	
	public static void main(String[] args) {
	String st1="WELCOME TO";
	System.*out*.println("string 1 is :"+ st1);
	String st2="Java Programming Language";
	System.*out*.println("string 2 is :"+ st2);
	String st=st1.concat(st2);
	System.*out*.println("string concatenation : " + st);
	System.*out*.println("length of string "+ st2 +" is " +st2.length());
	System.*out*.println("Character at position 5 of string 2 : " + st2.charAt(5));
	System.*out*.println("Index of character 'O' in string 2 : " + st2.indexOf('o'));
	System.*out*.println("Compare To 'WELCOME': " + st1.compareTo("Welecome"));
	System.*out*.println("Contains sequence 'ing' in string 2 : " + st2.contains("ing"));
	System.*out*.println("EndsWith character 'O' in string 1 : " + st1.endsWith("O"));
	System.*out*.println("LowerCase of string 1 : " + st1.toLowerCase());

```
                    System.out.println("UpperCase of string 2 : " +
st2.toUpperCase());
                    System.out.println("Replace 'WELCOME TO': " +
st.replace("WELCOME TO", "HOW IS"));
                    // TODO Auto-generated method stub

          }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  Javadoc  Declaration  Console  Properties
<terminated> String_Manip [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.just
string 1 is :WELCOME TO
string 2 is :Java Programming Language
string concatenation : WELCOME TO Java Programming Language
length of string Java Programming Language is 25
Character at position 5 of string 2 : P
Index of character 'O' in string 2 : 7
Compare To 'WELCOME': -32
Contains sequence 'ing' in string 2 : true
EndsWith character 'O' in string 1 : false
LowerCase of string 1 : welcome to
UpperCase of string 2 : JAVA PROGRAMMING LANGUAGE
Replace 'WELCOME TO': HOW IS Java Programming Language
```

PROGRAM 9

AIM

Program to create a class for Employee having attributes eNo, eName eSalary. Read n employ information and Search for an employee given eNo, using the concept of Array of Objects.

ALGORITHM

Step 1: Start

Step 2: Search the ‘eNo’ attribute of the list of Employee Objects for the ‘eNo’ provided by the user.

Step 3: If user provided ‘eNo’ is found inside the Employee object list, display the details of the corresponding employee.

Step 4: Stop

PROGRAM CODE

Employee.java

```
import java.util.Scanner;
public class Employee {
	int eNo;
	String eName;
	double eSalary;
	void getdata()
	{
		Scanner sc=new Scanner(System.in);

		System.out.println("Enter Employee Id ");
		eNo=sc.nextInt();
		System.out.println("Enter Employee Name");
		eName=sc.next();
		System.out.println("Enter Employee Salary");
		eSalary=sc.nextDouble();
	}
	void display()
	{
		System.out.println("Employee id is : "+ eNo);
		System.out.println("Employee name is : "+ eName);
		System.out.println("Employee salary is : "+ eSalary);
	}
	public static void main(String[] args) {
```

```
                Scanner sc1=new Scanner(System.in);
                int i,n,c,f=0;
                System.out.println("Enter the number of Employees");
                n=sc1.nextInt();
                Employee e[]=new Employee[n];
                for(i=0;i<n;i++) {
                        e[i]=new Employee();
                        e[i].getdata();
                }
                System.out.println("Employee details are :\n");
                for(i=0;i<n;i++) {
                        e[i].display();
                }
                System.out.println("\nEnter id of Employee to search: ");
                c=sc1.nextInt();
                for(i=0;i<n;i++) {
                if(c==e[i].eNo) {
                        f=1;
                        break;
                }
        }
        if(f==1) {
                System.out.println("Details of employee is ");
                e[i].display();
        }
        else
                System.out.println("Employee Id is Invalid");

        }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Employee [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.o
Enter the number of Employees
3
Enter Employee Id
1010
Enter Employee Name
David
Enter Employee Salary
56000
Enter Employee Id
1015
Enter Employee Name
Rahul
Enter Employee Salary
48000
Enter Employee Id
1023
Enter Employee Name
Aravind
Enter Employee Salary
47000
Employee details are :

Employee id is : 1010
Employee name is : David
Employee salary is : 56000.0
Employee id is : 1015
Employee name is : Rahul
Employee salary is : 48000.0
Employee id is : 1023
Employee name is : Aravind
Employee salary is : 47000.0

Enter id of Employee to search:
1015
Details of employee is
Employee id is : 1015
Employee name is : Rahul
Employee salary is : 48000.0
```

PROGRAM 10

AIM

Area of different shapes using overloaded functions

ALGORITHM

Step 1: Start

Step 2: Create a class area with function named area() with different numbers of paramters to achieve overloading and thus find area of square, circle, rectangle and trianle.

Step 3: Create object for the class and call appropriate functions.

Step 4: Stop

PROGRAM CODE

Shapes_Area.java

```
import java.util.Scanner;


public class Shapes_Area {
	void area (int side) {
		System.out.println("area of square is "+ side*side);
	}
	void area(float radius) {
		System.out.println("Area if circle is "+(3.14*radius*radius));
	}
	void area(int length, int breadth) {
		System.out.println("Area of rectanlge is "+length*breadth);
	}
	void area (float heigth, float width) {
		System.out.println("Area of triangle is
"+(0.5*heigth*width));
	}

	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		Shapes_Area ar=new Shapes_Area();
		System.out.println("Enter the side of square ");
		int side=sc.nextInt();
		System.out.println("Enter the radius of circle ");
		Float radius=sc.nextFloat();
		System.out.println("Enter the length and breadth of
rectangle");
```

```
                    int length=sc.nextInt();
                    int breadth=sc.nextInt();
                    System.out.println("Enter the heigth and width of triangle");
                    Float height=sc.nextFloat();
                    Float width=sc.nextFloat();
                    ar.area(side);
                    ar.area(radius);
                    ar.area(length,breadth);
                    ar.area(height,width);
                    // TODO Auto-generated method stub

            }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Shapes_Area [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.just
Enter the side of square
25
Enter the radius of circle
20
Enter the length and breadth of rectangle
15 20
Enter the heigth and width of triangle
20 25
area of square is 625
Area if circle is 1256.0
Area of rectanlge is 300
Area of triangle is 250.0
```

PROGRAM 11

AIM

Create a class 'Employees' with data members Empid, Name, Salary, Address and constructors to initialize the data members. Create another class 'Teacher' that inherit the properties of class employee and contain its own data members department, Subjects taught and constructors to initialize these data members and also include display function to display all the data members. Use array of objects to display details of N teachers.

ALGORITHM

Step 1: Start

Step 2: Create a class named 'Employee' with data members eid, name, sal, Address and a constructor Employee() to initialize them.

Step 3: Create a class named 'Teacher' which is derived from Employee, with data members dept, sub and a function display() to display details and a constructor Teacher().

Step 4: Create an array of objects of Teacher type, read details and display them

Step 5: Stop

PROGRAM CODE

Employees.java

```
import java.util.Scanner;

public class Employees {
        int Empid;
        String Name;
        double Salary;
        String Address;
        Scanner Empl=new Scanner(System.in);

        public Employees()
        {
                System.out.println("Enter Employee id: ");
                Empid=Empl.nextInt();
                System.out.println("Enter Name: ");
                Name=Empl.next();
                System.out.println("Enter Salary: ");
                Salary=Empl.nextDouble();
                System.out.println("Enter Address: ");
                Address=Empl.next();
```

```
        }
        public static void main(String[] args) {
                int i,n;
                Scanner input=new Scanner(System.in);
                System.out.println("Enter number of Employees to add: ");
            n=input.nextInt();
                Teacher obj[]=new Teacher[n];
                for(i=0;i<n;i++)
                {
                        obj[i]=new Teacher();
                }
                for(i=0;i<n;i++)
                {
                        obj[i].display();
                }

        }
}
class Teacher extends Employees
{
        String department;
        String subject;
        Scanner teach=new Scanner(System.in);

        public Teacher()
        {
                System.out.println("Enter department: ");
                department=teach.next();
                System.out.println("Enter Subject: ");
                subject=teach.next();
        }

        void display()
        {
                System.out.println("Enter Employee Details");
                System.out.println("Enter Employee id: "+Empid);
                System.out.println("Enter Employee Name: "+Name);
                System.out.println("Enter Employee Salary: "+Salary);
                System.out.println("Enter Employee Address: "+Address);
                System.out.println("Enter Teaching Department:
"+department);
                System.out.println("Enter Teaching Subject: "+subject);
        }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Employees [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.c
Enter number of Employees to add:
2
Enter Employee id:
101
Enter Name:
Rahul
Enter Salary:
35000
Enter Address:
Trivandrum
Enter department:
Chemistry
Enter Subject:
Bio-chemistry
Enter Employee id:
102
Enter Name:
Arun
Enter Salary:
40000
Enter Address:
Kollam
Enter department:
Biology
Enter Subject:
Zoology
Enter Employee Details
Enter Employee id: 101
Enter Employee Name: Rahul
Enter Employee Salary: 35000.0
Enter Employee Address: Trivandrum
Enter Teaching Department: Chemistry
Enter Teaching Subject: Bio-chemistry
Enter Employee Details
Enter Employee id: 102
Enter Employee Name: Arun
Enter Employee Salary: 40000.0
Enter Employee Address: Kollam
Enter Teaching Department: Biology
Enter Teaching Subject: Zoology
```

PROGRAM 12

AIM

Create a class 'Person' with data members Name, Gender, Address, Age and a constructor to initialize the data members and another class 'Employee' that inherits the properties of class Person and also contains its own data members like Empid, Company_name, Qualification, Salary and its own constructor. Create another class 'Teacher' that inherits the properties of class Employee and contains its own data members like Subject, Department, Teacherid and also contain constructors and methods to display the data members. Use array of objects to display details of N teachers.

ALGORITHM

Step 1: Start

Step 2: Create a class named 'Person' with data members name, gender, address and age & a constructor to initialize them.

Step 3: Create a class named 'Employee' which is derived from Person, with data members Empid, com_name, Emp_quali and Emp_salary & a constructor Employee() to initialize them.

Step 4: Create class named 'Teach" which is derived from Employee, with data members TeacherId , department, subject & a constructor Teach() to initilize members and a function named display() to display details.

Step 4: Create an array of objects of type Teach and display details.

Step 5: Stop

PROGRAM CODE

Inheritance2.java	import java.util.*; class Person { String Name; String Gender; String Address; int Age; Scanner pe=new Scanner(System.*in*);

```
        public Person() {
                System.out.println("\nEnter Person Name: ");
                Name=pe.next();
                System.out.println("Enter Gender: ");
                Gender=pe.next();
                System.out.println("Enter Address: ");
                Address=pe.next();
                System.out.println("Enter Age: ");
                Age=pe.nextInt();
        }
}
class Employeee extends Person
{
        int Empid;
        String Com_name;
        String Emp_quali;
        double Emp_salary;
        Scanner emp=new Scanner(System.in);

        public Employeee()
        {
                System.out.println("Enter Employee Id: ");
                Empid=emp.nextInt();
                System.out.println("Enter Company Name: ");
                Com_name=emp.next();
                System.out.println("Enter Employee Qualification: ");
                Emp_quali=emp.next();
                System.out.println("Enter Employee Salary: ");
                Emp_salary=emp.nextDouble();
        }
}
class teachers extends Employeee
{
        int TeacherId;
        String department;
        String subject;
        Scanner te=new Scanner(System.in);

        public teachers()
        {
                System.out.println("Enter Department: ");
                department=te.next();
                System.out.println("Enter Teacher Id: ");
                TeacherId=te.nextInt();
                System.out.println("Enter Subject: ");
                subject=te.next();
        }

        public void display()
        {
                System.out.println("\nDetails of Teacher Id with
"+TeacherId);
                System.out.println("Name : "+Name);
                System.out.println("Gender : "+ Gender);
                System.out.println("Address: "+Address);
```

```
                System.out.println("Age: "+Age);
                System.out.println("Employee id: "+Empid);
                System.out.println("Company Name: "+Com_name);
                System.out.println("Qualification: "+Emp_quali);
                System.out.println("Salary: "+Emp_salary);
                System.out.println("Department: "+department);
                System.out.println("Subject: "+subject);
                System.out.println("\n");

        }
}
public class Inheritance2
{
        public static void main(String[] args) {
                int n;
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter number of Teachers: ");
                n=sc.nextInt();

                teachers obj[]= new teachers[n];
                for(int i=0;i<n;i++)
                {
                obj[i]=new teachers();
                }
                for(int i=0;i<n;i++)
                {
                System.out.println("\nDetails of Employees: "+(i+1));
                obj[i].display();
                sc.close();
                }
        }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Inheritance2 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj
Enter number of Teachers:
2

Enter Person Name:
Aravind
Enter Gender:
M
Enter Address:
Trivandrum
Enter Age:
27
Enter Employee Id:
1005
Enter Company Name:
Corodova
Enter Employee Qualification:
Msc
Enter Employee Salary:
40000
Enter Department:
Physics
Enter Teacher Id:
205
Enter Subject:
Physics
```

```
Enter Person Name:
Arun
Enter Gender:
M
Enter Address:
Kollam
Enter Age:
29
Enter Employee Id:
1009
Enter Company Name:
Corodova
Enter Employee Qualification:
M.Phil
Enter Employee Salary:
56000
Enter Department:
Chemistry
Enter Teacher Id:
210
Enter Subject:
Chemicals
```

```
Details of Employees: 1

Details of Teacher Id with 205
Name : Aravind
Gender : M
Address: Trivandrum
Age: 27
Employee id: 1005
Company Name: Corodova
Qualification: Msc
Salary: 40000.0
Department: Physics
Subject: Physics



Details of Employees: 2

Details of Teacher Id with 210
Name : Arun
Gender : M
Address: Kollam
Age: 29
Employee id: 1009
Company Name: Corodova
Qualification: M.Phil
Salary: 56000.0
Department: Chemistry
Subject: Chemicals
```

PROGRAM 13

AIM

Write a program has class Publisher, Book, Literature and Fiction. Read the information and print the details of books from either the category, using inheritance.

ALGORITHM

Step 1: Start

Step 2: Create a class named 'Publisher' with data members p_name, p_year; a constructor named Publisher().

Step 3: Create a class named 'Book' which is derived 'Publisher' with data members b_name,b_author, b_price; a constructor named Book().

Step 4: Create a class named 'literature' which is derived from Book with data member page; a constructor; a function display() to display details.

Step 5: Create a class named 'fiction' which is derived from Book with data member page a constructor; a function display() to print details.

Step 6: Print a menu defining the type of genres; if literature create an object of literature type and object of type fiction if fiction is chosen.

Step 7: Stop

PROGRAM CODE

Inheritance3.java	

```
import java.util.Scanner;

class publisher {
        String p_name;
        int p_year;
        Scanner sc=new Scanner(System.in);

        publisher()
        {
                System.out.println("Enter Publisher name");
                p_name=sc.next();
                System.out.println("Enter the Year of Publication");
                p_year=sc.nextInt();
        }
}
```

```
class book extends publisher {
	String b_name,b_author;
	int b_price;
	Scanner sc=new Scanner(System.in);

	book() {
		System.out.println("Enter Book name");
		b_name=sc.next();
		System.out.println("Enter author");
		b_author=sc.next();
		System.out.println("Enter price");
		b_price=sc.nextInt();
	}
}

class literature extends book {
	int page;
	Scanner sc=new Scanner(System.in);

	literature() {
		System.out.println("Enter number of pages: ");
		page=sc.nextInt();
	}
	void display()
	{
		System.out.println("........LITERATURE BOOKS
ARE........");
		System.out.println("Publisher name is "+p_name);
		System.out.println("Published year is "+p_year);
		System.out.println("Book name is "+b_name);
		System.out.println("Autho name is "+b_author);
		System.out.println("Price is "+b_price);

	}
}


class fictions extends book {
	int page;
	Scanner sc=new Scanner(System.in);

	fictions() {
		System.out.println("Enter number of pages");
		page=sc.nextInt();
	}
	void display()
	{
		System.out.println(".........FICTION BOOKS ARE.........");
		System.out.println("Publisher name is "+p_name);
		System.out.println("Published year is "+p_year);
		System.out.println("Book name is "+b_name);
		System.out.println("Autho name is "+b_author);
		System.out.println("Price is "+b_price);
```

```
        }
}
public class Inheritance3 {

        public static void main(String[] args) {
                int n,m,c;
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter number of literatures books");
                n=sc.nextInt();
                literature l[]=new literature[n];
                for(int i=0;i<n;i++) {
                        l[i]=new literature();
                }
                System.out.println("Enter number of fictions books");
                m=sc.nextInt();
                fictions f[]=new fictions[m];
                for(int i=0;i<m;i++) {
                        f[i]=new fictions();
                }
                System.out.println("Enter your Choice
\n1:LITERATURE\n2:FICTION");
                c=sc.nextInt();
                if(c==1) {
                        for(int i=0;i<n;i++) {
                                l[i].display();
                        }
                }
                else if(c==2) {
                        for(int i=0;i<n;i++) {
                                f[i].display();
            }
        }
                else
                        System.out.println("Wrong choice");
         sc.close();
  }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Inheritance3 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj
Enter number of literatures books
2
Enter Publisher name
AKG_Publishers
Enter the Year of Publication
2002
Enter Book name
Thermodynamics
Enter author
Sharafudheen_S
Enter price
450
Enter number of pages:
325
Enter Publisher name
E_Books
Enter the Year of Publication
2012
Enter Book name
Ways_Artificial_Intelligence
Enter author
John_Ebraham
Enter price
550
Enter number of pages:
375
```

```
Enter number of fictions books
2
Enter Publisher name
Leadstart_Publishing
Enter the Year of Publication
2008
Enter Book name
Asura
Enter author
Anand_Neelakanthan
Enter price
450
Enter number of pages
300
Enter Publisher name
Fingerprint_Publishing
Enter the Year of Publication
2014
Enter Book name
Seven_Uncommoners
Enter author
Ridhima_Verma
Enter price
505
Enter number of pages
435
```

```
Enter your Choice
1:LITERATURE
2:FICTION
1
........LITERATURE BOOKS ARE........
Publisher name is AKG_Publishers
Published year is 2002
Book name is Thermodynamics
Autho name is Sharafudheen_S
Price is 450
........LITERATURE BOOKS ARE........
Publisher name is E_Books
Published year is 2012
Book name is Ways_Artificial_Intelligence
Autho name is John_Ebraham
Price is 550
```

PROGRAM 14

AIM

Create classes Student and Sports. Create another class Result inherited from Student and Sports. Display the academic and sports score of a student.

ALGORITHM

Step 1: Start

Step 2: Create a class named ‘Student’ with data members as name,roll_no and subject names; a constructor.

Step 3: Create a class named ‘Sports’ which is derived ‘Student’ with data members football and cricket; a constructor.

Step 4: Create a class named ‘Result’ which is derived from ‘Sports’ with data member total which shows the total score get in academics; a function display() to display details.

Step 5: Create an object of type Student, Sports and Result, and display the details.

Step 6: Stop

PROGRAM CODE

Final_score_card.java	

```
import java.util.Scanner;

class student {
	String Name;
	int Roll_No,English,Maths,Science,Social;
	Scanner sc=new Scanner(System.in);
	public student() {
		System.out.println("Enter the name of the student");
		Name=sc.next();
		System.out.println("Enter Student Roll No: ");
		Roll_No=sc.nextInt();
		System.out.println("Enter the mark of English: ");
		English=sc.nextInt();
		System.out.println("Enter the mark of Maths: ");
		Maths=sc.nextInt();
		System.out.println("Enter the mark of Science: ");
		Science=sc.nextInt();
		System.out.println("Enter the mark of Social: ");
		Social=sc.nextInt();
	}
```

```
}
class sports extends student{
        String Football,Cricket;
        public sports() {
                System.out.println("Enter the grade in Football");
                Football=sc.next();
                System.out.println("Enter the grade in Cricket");
                Cricket=sc.next();
        }
}
class result extends sports{
        public result() {

        }
        int Total=English+Maths+Science+Social;
        void display(){
                System.out.println("------Score Card of Student "+Name+"------");
                System.out.println("Subjects ");
                System.out.println("English out of 100: "+English);
                System.out.println("Maths out of 100: "+Maths);
                System.out.println("Science out of 100: "+Science);
                System.out.println("Social out of 100: "+Social);
                System.out.println("Total Scored in Academics out of 400:
"+Total);
                System.out.println("--Sports Grades-- ");
                System.out.println("Football Grade ---> "+Football);
                System.out.println("Cricket Grade --->"+Cricket);
        }
}
public class Final_score_card {

        public static void main(String[] args) {

                result obj = new result();
                obj.display();

        }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Final_score_card [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.
Enter the name of the student
Arjun
Enter Student Roll No:
220
Enter the mark of English:
87
Enter the mark of Maths:
80
Enter the mark of Science:
89
Enter the mark of Social:
85
Enter the grade in Football
A
Enter the grade in Cricket
A+
------Score Card of Student Arjun------
Subjects
English out of 100: 87
Maths out of 100: 80
Science out of 100: 89
Social out of 100: 85
Total Scored in Academics out of 400: 341
--Sports Grades--
Football Grade ---> A
Cricket Grade --->A+
```

PROGRAM 15

AIM

Create an interface having prototypes of functions area() and perimeter(). Create two classes Circle and Rectangle which implements the above interface. Create a menu driven program to find area and perimeter of objects.

ALGORITHM

Step 1: Start

Step 2: Create an interface ‘shapes’ with 2 functions area() and perimeter().

Step 3: Create a class named circle to implement the functions of interface to find area and perimeter of circle.

Step 4: Create a class named rectangle to implement the functions of interface to find area and perimeter of rectangle.

Step 5: Create objects for both these classes and call functions area() and perimeter() to display the same.

Step 6: Stop.

PROGRAM CODE

Objects.java

```
import java.util.Scanner;

interface shapes
{
        void area();
        void perimeter();
}

class circle implements shapes
{
        int r ;
        double pi = 3.14,area,perimeter;

  public circle()
        {
        Scanner sc = new Scanner(System.in);
                System.out.println("Enter Radius of circle: ");
                r = sc.nextInt();
```

```
        }
        public void area()
        {

                area = pi * r  *r;
                System.out.println("Area of circle with radius "+r+" is " + area);

        }
        public void perimeter()
        {
                perimeter = 2 * pi * r;
                System.out.println("Perimeter of circle with radius "+r+" is " +
perimeter);
        }
}
class rectangle implements shapes
{
  int l ,b;
  int area,perimeter;

  public rectangle()
  {
            Scanner sc = new Scanner(System.in);
                System.out.println("Enter Length of rectangle: ");
                l = sc.nextInt();
                System.out.println("Enter Breadth of rectangle: ");
                b = sc.nextInt();
  }
   public void area()
   {

                area = l *b;
                System.out.println("Area of rectangle is: " + area);

   }
   public void perimeter()
   {
        perimeter = 2 *(l+b);
        System.out.println("Perimeter of rectangle is: " + perimeter);
   }
}
public class Objects {

        public static void main(String[] args) {
                {
                        int ch1,ch2;
                        Scanner sc = new Scanner(System.in);
                        System.out.println("Select a shape \n 1.Circle \n
2.Rectangle");
                        System.out.println("Enter Your Choice : ");
                        ch1 = sc.nextInt();
                        switch(ch1)
                        {
                        case 1 : circle obj1 = new circle();
                                System.out.println("Find \n1.Area \n2.Perimeter");
```

```
                                    System.out.println("Enter Your Choice : ");
                                    ch2 = sc.nextInt();
                                     switch(ch2)
                                     {
                                       case 1 : obj1.area();
                                                break;
                                        case 2 : obj1.perimeter();
                                                 break;
                                        default : System.out.println("Invalid choice");
                                      }
                                     break;

                              case 2 :  rectangle obj2 = new rectangle();
                                         System.out.println("Find \n1.Area
\n2.Perimeter");
                                         System.out.println("Enter Your Choice : ");
                                         ch2 = sc.nextInt();
                                        switch(ch2)
                                        {
                                          case 1 : obj2.area();
                                                   break;
                                          case 2 : obj2.perimeter();
                                                   break;
                                          default : System.out.println("Invalid choice");
                                         }
                                        break;
                                        default : System.out.println("Invalid choice");

                                }
                        // TODO Auto-generated method stub

                }

  }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Objects [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.ope
Select a shape
 1.Circle
 2.Rectangle
Enter Your Choice :
1
Enter Radius of circle:
25
Find
1.Area
2.Perimeter
Enter Your Choice :
1
Area of circle with radius 25 is 1962.5
```

PROGRAM 16

AIM

Prepare bill with the given format using calculate method from interface.

Order No.

Date:

Product Id	Name	Quantity	unit price	Total
101	A	2	25	50
102	B	1	100	100
			Net. Amount	150.

ALGORITHM

Step 1: Start

Step 2: Create an interface 'Bill' with function total().

Step 3: Create a class named 'product' that implements the interface Bill with data members Order_no, P_Id, P_Name, Qty, Unit_price, Total and Net_Amt.

Step 4: create methods Product_detls() and Order_No() getting details from the user as above format. A function display() to print details.

Step 4: Create an object of type billcalc to print the bill.

Step 5: Stop

PROGRAM CODE

Bill_Receipt.java	import java.util.Scanner;
	interface Bill
	{
	void total();
	}
	class Product implements Bill
	{

```
        int Order_No;
        int P_Id;
        String P_Name;
        float Qty;
        float Unit_Price;
        float Total;
        float Net_Amt=0;
        Scanner sc=new Scanner(System.in);
        public void Product_Detls()
        {
                System.out.println("Enter the Product Id: ");
                P_Id=sc.nextInt();
                System.out.println("Enter the Product Name: ");
                P_Name=sc.next();
                System.out.println("Enter the Quantity: ");
                Qty=sc.nextFloat();
                System.out.println("Enter Unit Price: ");
                Unit_Price=sc.nextFloat();
        }
        public void total()
        {
                Total=Qty*Unit_Price;
                System.out.println("
"+P_Id+"\t\t"+P_Name+"\t\t"+Qty+"\t\t"+Unit_Price+"\t\t"+Total+"\n");
        }
        public void Order_No()
        {
                System.out.println("\nEnter the Order No: ");
                Order_No=sc.nextInt();
        }
        public void display()
        {
                System.out.println("\n-------------Bill Receipt------------\n");
                System.out.println("\nOrder No." + Order_No);
                System.out.println("\nDate : " + java.time.LocalDate.now());
                System.out.println("\nProduct Id\t Name\t\tQuantity\tunit
price\tTotal");

        System.out.print("_____________________________________________________
_____________________________\n");
        }
}
public class Bill_Receipt {

        public static void main(String[] args) {
                int n;
                int Net_Amt=0;
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter number of Items Purchased: ");
                n=sc.nextInt();
                Product Pro= new Product();
                Product obj[]= new Product[n];
                Pro.Order_No();
                for(int i=0;i<n;i++)
                {
```

```
                    System.out.println("\nEnter Details of Product"+(i+1));
                    obj[i]=new Product();
                    obj[i].Product_Detls();
                    }

                Pro.display();
                    for(int i=0;i<n;i++)
                    {
                      obj[i].total();
                      Net_Amt+=obj[i].Total;
                    }

            System.out.print("_____________________________________________
_____________________________");
                    System.out.println("\n\t\t\t\t\t\tNet. Amount\t  "+Net_Amt);
                    sc.close();

            }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> Bill_Receipt [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.openjdk

Enter the Order No:
12312

Enter Details of Product1
Enter the Product Id:
101
Enter the Product Name:
A
Enter the Quantity:
2
Enter Unit Price:
25

Enter Details of Product2
Enter the Product Id:
102
Enter the Product Name:
B
Enter the Quantity:
1
Enter Unit Price:
100

-------------Bill Receipt------------


Order No.12312

Date : 2021-09-09

Product Id       Name          Quantity        unit price      Total
_____________________________________________________________________
  101            A             2.0             25.0            50.0

  102            B             1.0             100.0           100.0


_____________________________________________________________________
                                               Net. Amount       150
```

PROGRAM 17

AIM

Create a Graphics package that has classes and interfaces for figures Rectangle, Triangle, Square and Circle. Test the package by finding the area of these figures.

ALGORITHM

Step 1: Start

Step 2: To create a package named graphics, create a folder of the same name in the directory. Here inside that we have another module named calculate

Step 3: Inside the graphics folder, create modules for finding the areas of rectangle, circle, triangle and square.

Step 4: Outside the graphics folder, write a program to access the modules mention above and print the output

Step 5: Stop

PROGRAM CODE

Shapes.java	package Graphics;
	
	interface Result
	{
	void areaRectangle(float len,float br);
	void areaTriangle(float ba,float hei);
	void areaSquare(float side);
	void areaCircle(float r);
	
	}
	
	public class Shapes implements Result
	{
	public void areaRectangle(float len,float br)
	{
	System.*out*.println("Area of Rectangle:"+(len*br));
	}
	public void areaTriangle(float ba,float hei)
	{
	System.*out*.println("Area of the triangle =="+(0.5*ba*hei));

```
    }
    public void areaSquare(float side)
    {
                System.out.println("Area of the square ="+(side*side));
    }
    public void areaCircle(float r)
    {
       System.out.println("Area of Circle ="+3.14*r*r);
    }

    public static void main(String[] args) {

    }
}
```

Areaqn1.java

```
import Graphics.Shapes;
public class Areaqn1
{
        public static void main(String[] args)
        {
                int ch;
                float Rec_len,Rec_br;
                float Tri_ba,Tri_hei;
                float side;
                float r;
                Shapes obj=new Shapes();
                Scanner sc = new Scanner(System.in);

                        System.out.println("Select a shape \n 1.Rectangle \n
2.Triangle \n 3.Square \n 4.Circle\n");
                        System.out.println("Enter Your Choice : ");
                        ch = sc.nextInt();
                        switch(ch)
                        {
                                case 1 : System.out.println("Enter the
Length:");
                         Rec_len = sc.nextFloat();
                         System.out.println("Enter the Breadth:");
                         Rec_br = sc.nextFloat();

obj.areaRectangle(Rec_len,Rec_br);
                                        break;
                                case 2 : System.out.println("Enter the
Base:");
                                                Tri_ba = sc.nextFloat();
                                                System.out.println("Enter the
Height:");
                                                Tri_hei = sc.nextFloat();

obj.areaTriangle(Tri_ba,Tri_hei);
```

```
                                        break;
                                    case 3 : System.out.println("Enter the Side:");
                                                side = sc.nextFloat();
                                                obj.areaSquare(side);
                                        break;
                                    case 4 : System.out.println("Enter the
Radius:");
                                                r = sc.nextFloat();
                                                obj.areaCircle(r);
                                        break;
                                    case 5 : System.exit(0);
                                        break;
                                    default : System.out.println("Invalid
choice");
                            }
                            sc.close();

                    // TODO Auto-generated method stub
            }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console
<terminated> Area [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse.justj.openjdk
Select a shape
 1.Rectangle
 2.Triangle
 3.Square
 4.Circle

Enter Your Choice :
2
Enter the Base:
60
Enter the Height:
55
Area of the triangle  =1650.0
```

PROGRAM 18

AIM

Create an Arithmetic package that has classes and interfaces for the 4 basic arithmetic operations. Test the package by implementing all operations on two given numbers

ALGORITHM

Step 1: Start

Step 2: To create a package named Arithmetic, create a folder of the same name in the directory. Here inside that we have another module named operation

Step 3: Inside Arithmetic package, create modules to perform addition, subtraction, multiplication and division of 2 numbers.

Step 4: Outside the folder, write another program that access the above module and print the output.

Step 5: Stop

PROGRAM CODE

Operations.java	package Arithmetic;
	interface Calculation
	{
	void Multiplication(int a,int b);
	void Division(float a,float b);
	void Addition(int a,int b);
	void Subtraction(int a,int b);
	
	}
	public class Operations implements Calculation
	{
	public void Multiplication(int a,int b)
	{
	System.out.println(a+"x"+b+"="+(a*b));
	}
	public void Division(float a,float b)
	{
	System.out.println(a+"/"+b+"="+(a/b));
	}
	public void Addition(int a,int b)

```
            {
                        System.out.println(a+"+"+b+"="+(a+b));
            }
            public void Subtraction(int a,int b)
            {
               System.out.println(a+"-"+b+"="+(a-b));
            }

        public static void main(String[] args) {
                // TODO Auto-generated method stub

        }

}
```

Inputsqn2.java

```
import java.util.Scanner;

import Arithmetic.Operations;
public class Inputsqn2 {

        public static void main(String[] args) {
                int a,b;
                int ch;
                Operations obj=new Operations();
                Scanner sc=new Scanner(System.in);
                System.out.println("Enter the 2 numbers:");
            a=sc.nextInt();
            b=sc.nextInt();

System.out.println("\n1.Multiplication\n2.Division\n3.Addition\n4.Substra
ction\n");
            System.out.println("Enter the choice:\n");
            ch = sc.nextInt();
            switch(ch)
      {
        case 1:obj.Multiplication(a,b);
              break;
        case 2:obj.Division(a,b);
              break;
        case 3:obj.Addition(a,b);
              break;
        case 4:obj.Subtraction(a,b);
              break;
        default:System.out.println("Invalid choice");
      }
            sc.close();
                // TODO Auto-generated method stub

        }
```

	}

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console
<terminated> Inputs [Java Application] C:\Users\brahm\.p2\pool\plugins\org
Enter the 2 numbers:
45
9

1.Multiplication
2.Division
3.Addition
4.Substraction

Enter the choice:

4
45-9=36
```

PROGRAM 19

AIM

Write a user defined exception class to authenticate the user name and password.

ALGORITHM

Step 1: Start

Step 2: Create a class named UsernameException that inherits Exception class with a constructor that

calls Exception class constructor and pass error meaasage.

Step 3: Create a class named PasswordException that inherits Exception class with a

constructor that calls Exception class constructor and pass error meaasage.

Step 4: Inside the main(), Read the username and password.

Step 5: Inside the try block, we throw UsernamException and PasswordException with

appropriate message if any of the conditon is true:

- If username is empty
- If password is empty
- If password doesnt contain special charecters
- If username length is less than 6
- If password is not string enough

Step 6: Inside the catch block with parameter UsernameException's object, print

"USERNAME EXCEPTION OCCURED"

Step 7: Inside the catch block with parameter PasswordException's object, print

"PASSWORD EXCEPTION OCCURED"

Step 8: Stop

PROGRAM CODE

Authenticate.java	import java.util.Scanner; class UsernameException extends Exception{ /** *

```
	 */
	private static final long serialVersionUID = 1L;

	public UsernameException(String U_name) {
	  super(U_name);
	}
	}
class PasswordException extends Exception{
	/**
	 *
	 */
	private static final long serialVersionUID = 1L;

	public PasswordException(String P_word) {
		super(P_word);
	}
}
public class Authenticate {

	public static void main(String[] args) {
		Scanner sc= new Scanner(System.in);
		String uname,pwd;
		int length;
		System.out.println("Enter Username: ");
		uname=sc.nextLine();
		length=uname.length();
		System.out.println("Enter Password: ");
		pwd=sc.nextLine();

		try {
			if(length<8)
				throw new
UsernameException("Username must greater than 8 charecters");
			else if(!pwd.equals("Brahman@123"))
				throw new
PasswordException("Incorrect Password\n Type Correct one");
			else
				System.out.println("Successfully Logged
in.....");
			}

		catch(UsernameException u) {
			System.out.println("Exception Occurred. . "+u);
		}
		catch(PasswordException p) {
			System.out.println("Exception Occurred. . "+p);
		}
		sc.close();
		// TODO Auto-generated method stub
```

```
        }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
<terminated> Authenticate [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jr
Enter Username:
brahma
Enter Password:
Brahman@123
Exception Occurred. . UsernameException: Username must greater than 8 charecters
```

```
<terminated> Authenticate [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot
Enter Username:
Brahmaduttan
Enter Password:

Exception Occurred. . PasswordException: Incorrect Password
 Type Correct one
```

```
<terminated> Authenticate [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse.justj.openjdk.hc
Enter Username:
Brahmaduttan
Enter Password:
Brahman@123
Successfully Logged in.....
```

PROGRAM 20

AIM

Find the average of N positive integers, raising a user defined exception for each negative input.

ALGORITHM

Step 1: Start

Step 2: Create a class named NegativeIntException that inherits Exception class with a constructor inside which we call the Exception class constructor and pass error meaasage.

Step 3: Inside the main(), Read the limit of array

Step 4: Inside the try block,read the array and check if any element is less than 0

Step 5: If true, throw NegException with appropriate message.

Step 6: Calculate the average of the array and print it

Step 7: Inside the catch exception, Print “NEGETIVE EXCEPTION OCCURED”

Step 8: Stop

PROGRAM CODE

AvgofPostive.java	import java.util.Scanner;
	class NegativeIntException extends Exception
	{
	/**
	*
	*/
	private static final long serialVersionUID = 1L;
	
	public NegativeIntException(String s)
	{
	super(s);
	}
	}
	public class AvgofPositive {
	
	public static void main(String[] args) {

```
                int n,i;
                int sum=0;
        int num[];
        float avg,count=0;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the total number to find
average:");
        n = sc.nextInt();
        num = new int[n];
        try {
                System.out.println("Enter the numbers:");
                  for(i=0;i<n;i++)
                  {
                    num[i] = sc.nextInt();
                  }
                  for(i=0;i<n;i++)
                  {
                                if(num[i]<0)
                                {
                                        throw new
NegativeIntException("Entered numbers must positive");
                                }
                                else
                                {
                                     sum = sum + num[i];
                                     count++;
                                }
                  }
                  avg=sum/count;
                  System.out.println("Average :"+avg);
        }
         catch(NegativeIntException e) {
                System.out.println("Exception Occurred..... "+e);
      }

                // TODO Auto-generated method stub
      }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
<terminated> AvgofPositive [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspo
Enter the total number to find average:
5
Enter the numbers:
10
45
85
-50
25
Exception Occurred..... NegativeIntException: Entered numbers must positive
```

```
<terminated> AvgofPositive [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse.justj.openj
Enter the total number to find average:
5
Enter the numbers:
45
63
48
20
41
Average :43.4
```

PROGRAM 21

AIM

Define 2 classes; one for generating multiplication table of 5 and other for displaying first N prime numbers. Implement using threads. (Thread class)

ALGORITHM

Step 1: Start

Step 2: Create a class named Multi_Thread that inherits Thread class with member function as run()

Step 3: Inside run(), Print the multiplication table for 5

Step 4: Create a class named prime that inherits Thread class with memebr function run()

Step 5: Inside run(), Print the prime numbers upto the limit of user's choice

Step 6: Inised the main(), create an object for the classes and call start() using each object

Step 7: Stop

PROGRAM CODE

MultiThread.java	import java.util.Scanner;
	public class MultiThread {
	public static void main(String[] args)throws InterruptedException
	{
	ThreadMul a=new ThreadMul();
	a.start();
	Thread.sleep(200);
	ThreadPrime b=new ThreadPrime();
	b.start();
	Thread.sleep(200);
	
	// TODO Auto-generated method stub
	
	}
	}
	class ThreadMul extends Thread
	{
	public void run()
	{

```
        int n=5;
        System.out.println("Multiplication table Of 5\n");
        System.out.println("*************************\n");
        for(int i=1;i<=10;i++)
        {

            System.out.println("\t"+n+"x"+i+"="+n*i);
        }
        System.out.println("\n*************************\n");
    }
}
class ThreadPrime extends Thread
{
            public void run()
    {
        int i,count,j,limit;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the limit to find the Prime Numbers:");
        limit = s.nextInt();
        System.out.println("Prime numbers between 1 and " + limit + "
are:");
        for(i=1;i<=limit;i++)
        {
            count=0;
            for(j=1;j<=i;j++)
            {
                    if(i%j==0)
                    {
                            count++;
                }
            }
            if(count==2)
            {
                    System.out.println(i);
            }
        }

    }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console
<terminated> MultiThread [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse.justj.openjdk
Multiplication table Of 5

*************************

        5x1=5
        5x2=10
        5x3=15
        5x4=20
        5x5=25
        5x6=30
        5x7=35
        5x8=40
        5x9=45
        5x10=50

*************************

Enter the limit to find the Prime Numbers:
25
Prime numbers between 1 and 25 are:
2
3
5
7
11
13
17
19
23
```

PROGRAM 22

AIM

Define 2 classes; one for generating Fibonacci numbers and other for displaying even numbers in a given range. Implement using threads. (Runnable Interface).

ALGORITHM

Step 1: Start

Step 2: Create a class named even that implements Runnable interface with function run()

Step 3: Inside run(), we read the limit for printing even numbers and print it using for loop.

Step 4: Create another calss fib that implements Runnable interface with function run().

Step 5: Inside run(), Initialise n1 as 0,n2 as 1 and n3 as 0.

Step 6: Check if n<0, if true, print "Enter a positive number" else goto step 7

Step 7: Repeat step8 to 11 until n3>n

Step 8: Print n1

Step 9: n3=n1+n2

Step 10: n1=n2

Step 11: n2=n3

Step 12: Create object e of even and create an object t1 of Thread with its parameterized constructor passing e as parameter

Step 13: Call start() using t1

Step 14: Do the same for class odd with Thread object t2 and call start() using t2

Step 15: Stop

PROGRAM CODE

Fibonacci_Even.java	import java.util.Scanner; class Fibonacci implements Runnable { int n,first,second,t; String str;

```
    public Fibonacci(int num)
    {
        n = num;
        first = 0;
        second = 1;
    }

    @Override
    public void run()
    {
        str = first+" "+second;
        for(int i=0;i<=n-3;i++)
        {
            t = first + second;
            first = second;
            second = t;
            str += " "+t;
        }
        System.out.println(str);
    }
}
class Even implements Runnable
{
    int n;
    String str;
    public Even(int n)
    {
        this.n = n;
        str = "";
    }
    @Override
    public void run()
    {
        for(int i=0;i<n;i=i+2)
            if(i%2==0)
            {
                str+=i+" ";
            }
        System.out.println(str);
    }

}
public class Fibonacci_Even {

            public static void main(String[] args) throws
InterruptedException {
        int n1,n2;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the range to see the Fibanocci
Series: ");
```

```
        n1 = sc.nextInt();
        Fibonacci fib = new Fibonacci(n1);
        Thread th = new Thread(fib);
        th.start();
        Thread.sleep(400);
        System.out.println("Enter the range of even numbers: ");
        n2 = sc.nextInt();
        Even e = new Even(n2);
        Thread th2 = new Thread(e);
        th2.start();
        sc.close();
                // TODO Auto-generated method stub

    }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
<terminated> Fibonacci_Even [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse.justj.openjdk
Enter the range to see the Fibanocci Series:
10
0 1 1 2 3 5 8 13 21 34
Enter the range of even numbers:
50
0 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48
```

PROGRAM 23

AIM

Producer/Consumer using ITC.

ALGORITHM

Step 1: Start

Step 2: In PC class (A class that has both produce and consume methods), a linked list of jobs and a capacity of the list is added to check that producer does not produce if the list is full.

Step 3: In Producer class, the value is initialized as 0.

Step 4: We have an infinite outer loop to insert values in the list. Inside this loop, we have a synchronized block so that only a producer or a consumer thread runs at a time. An inner loop is there before adding the jobs to list that checks if the job list is full, the producer thread gives up the intrinsic lock on PC and goes on the waiting state.

Step 5: If the list is empty, the control passes to below the loop and it adds a value in the list.

Step 6: In the Consumer class, we again have an infinite loop to extract a value from the list. Inside, we also have an inner loop which checks if the list is empty.

Step 7: If it is empty then we make the consumer thread give up the lock on PC and passes the control to producer thread for producing more jobs.

Step 8: If the list is not empty, we go round the loop and removes an item from the list.

Step 9: In both the methods, we use notify at the end of all statements. The reason is simple, once you have something in list, you can have the consumer thread consume it, or if you have consumed something, you can have the producer produce something.

Step 10: sleep() at the end of both methods just make the output of program run in step wise manner and not display everything all at once so that you can see what actually is happening in the program.

Step 11: Stop

PROGRAM CODE

prod_cons.java

```
import java.util.LinkedList;

public class prod_cons {
        public static void main(String[] args) throws InterruptedException
        {
                final PC pc = new PC();
                Thread t1 = new Thread(new Runnable()
                {
                  public void run()
                  {
                        try
                        {
                          pc.produce();
                        }
                        catch (InterruptedException e)
                        {
                      e.printStackTrace();
                        }
                  }
                });

            Thread t2 = new Thread(new Runnable()
            {
            public void run()
             {
              try
              {
                    pc.consume();
              }
              catch (InterruptedException e)
              {
                    e.printStackTrace();
              }
             }
            });
            t1.start();
            t2.start();
            t1.join();
            t2.join();
        }
        public static class PC
        {
                LinkedList<Integer> list = new LinkedList<>();
                int capacity = 2;

           public void produce() throws InterruptedException
           {
```

```
                int value = 0;
                while (true)
                {
                        synchronized (this)
                        {
                                while (list.size() == capacity)
                                        wait();
                                System.out.println("Producer
produced:"+ value);
                                list.add(value++);
                                notify();
                                Thread.sleep(1000);
                        }
                }
        }
        public void consume() throws InterruptedException
        {
                while (true)
                {
                        synchronized (this)
                        {
                                while (list.size() == 0)
                                        wait();
                                int val = list.removeFirst();
                                System.out.println("Consumer
consumed:"+ val);
                                notify();
                                Thread.sleep(1000);
                        }
                }
        }
    }
    }
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

Problems @ Javadoc Declaration Console Properties

prod_cons [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot

```
Producer produced:0
Producer produced:1
Consumer consumed:0
Consumer consumed:1
Producer produced:2
Producer produced:3
Consumer consumed:2
Consumer consumed:3
Producer produced:4
Producer produced:5
Consumer consumed:4
Consumer consumed:5
Producer produced:6
Consumer consumed:6
Producer produced:7
Producer produced:8
Consumer consumed:7
Consumer consumed:8
Producer produced:9
Producer produced:10
Consumer consumed:9
Consumer consumed:10
Producer produced:11
```

PROGRAM 24

AIM

Program to create a generic stack and do the Push and Pop operations.

ALGORITHM

Step 1: Start

Step 2: Create a class named stack with data members as a(an array),top(set as -1),ch,item,i; a function named menu()

Step 3: Inside menu(), give choices to push,pop and display the stack

Step 4: If the choice is 1, then check whether the stack is full, else add an element into the stack.

Step 5: If the choice is 2, then check whether the stack is empty, else delete an element into the stack.

Step 6: If the choice is 3, then check whether the stack is empty, else print all the elements in the stack.

Step 7: If the choice is greater than 4, then print “Invalid option”.

Step 8: Inside the main(), create an object of type stack and call the menu() function.

Step 9: Stop

PROGRAM CODE

Stack.java	import java.util.Scanner;
	
	class Operations {
	public void operation() {
	int N,El,ch,top=-1;
	int size;
	Scanner sc=new Scanner(System.in);
	System.out.println("Enter Stack size: ");
	N=sc.nextInt();
	int[] arr=new int[N];
	
	do {
	System.out.println("\n_____\n");

```
                        System.out.println("  Operations : \n1.Push \n2.Pop
\n3.Display \n4.Exit  ");
                        System.out.println("\n____________________\n");
                        ch=sc.nextInt();
                        size=N-1;
                        switch(ch) {

                        case 1: if(top==size)
                                        {

        System.out.println("Warning!!!!!!!.......Stack is Full.....");
                                        }
                                        else
                                        {
                                                System.out.println("Enter your
Element: ");
                                                El=sc.nextInt();
                                                top++;
                                                arr[top]=El;
                                        }
                                        break;

                        case 2: if(top==-1)
                                        {

        System.out.println("Warning!!!!!!!.......Stack is Empty.....");
                                        }
                                        else
                                        {
                                                System.out.println("Element
"+arr[top]+" removed");
                                                top--;
                                        }
                                        break;

                        case 3: if(top==-1)
                                        {

        System.out.println("Warning!!!!!!!.......Stack is Empty.....");
                                        }
                                        else
                                        {
                                                System.out.println("Entered
Stack: ");
                                                System.out.println("------");
                                                for(int i=top;i>=0;i--)
                                                {
                                                        System.out.println("
"+arr[i]);
```

```
                                                System.out.println("------
");
                                                        }
                                                }
                                                break;

                        case 4: System.exit(0);

                        default: System.out.println("Invalid choice!!!");

                        }
                }while(ch!=4);
                sc.close();
        }

}
public class Stack {

        public static void main(String[] args) {
                Operations op= new Operations();
                op.operation();
                // TODO Auto-generated method stub

        }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Stack [Java Application] C:\Users\brahm\.p2\pool\p
Enter Stack size:
3


_________________

  Operations :
1.Push
2.Pop
3.Display
4.Exit


_________________

1
Enter your Element:
3


_________________

  Operations :
1.Push
2.Pop
3.Display
4.Exit


_________________

1
Enter your Element:
6


_________________

  Operations :
1.Push
2.Pop
3.Display
4.Exit


_________________
```

```
Stack [Java Application] C:\Users\brahm\.p2\pool\plugir
_________________

1
Enter your Element:
5


_________________

  Operations :
1.Push
2.Pop
3.Display
4.Exit


_________________

3
Entered Stack:
------
  5
------
  6
------
  3
------


_________________

  Operations :
1.Push
2.Pop
3.Display
4.Exit


_________________

2
Element 5 removed


_________________
```

```
Stack [Java Application] C:\Users\brahm\.p2\
  Operations :
1.Push
2.Pop
3.Display
4.Exit

_________________

2
Element 5 removed

_________________

  Operations :
1.Push
2.Pop
3.Display
4.Exit

_________________

3
Entered Stack:
------
  6
------
  3
------

_________________

  Operations :
1.Push
2.Pop
3.Display
4.Exit

_________________
```

PROGRAM 25

AIM

Using generic method perform Bubble sort.

ALGORITHM

Step 1: Start

Step 2: Create a function named bubblesort(array)

Step 3: n<- length of array

Step 4: Intialize temp<-0

Step 5: i<-0

Step 6: Reapeat steps from to until i>n

Step 7: j<-1,repeat the steps from to until j>n-I

Step 8: check if array[i] >array[j], if true,swap them;else increment j

Step 9: Inside main () Initialize an array with elements and the print the same

Step 10: Call the function bubblesort() and pass the array as parameter

Step 11: Print the sorted array

Step 12: Stop

PROGRAM CODE

Bubble_sort.java	import java.util.Scanner;
	
	class Bubblesrt{
	void sort(int n,int arr[])
	{
	int i,j,temp;
	for(i=0;i<n;i++)
	{
	for(j=0;j<n-1;j++)
	{
	if(arr[j]>arr[j+1])
	{
	temp=arr[j];

```
                                        arr[j]=arr[j+1];
                                        arr[j+1]=temp;
                                    }
                                }
                            }
                        }
        void display(int n,int arr[])
        {
                int i;
                for(i=0;i<n;i++)
                  {
                        System.out.println(arr[i]+" ");
                  }
        }

}
public class Bubble_sort {

        public static void main(String[] args) {
                int n,i;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the No of Elements to sort: ");
                n=sc.nextInt();
                int arr[] =new int[n];
                System.out.println("Enter "+n+" Elements");
                for(i=0;i<n;i++)
                  {
                        arr[i]=sc.nextInt();
                  }
                sc.close();
                Bubblesrt bsrt= new Bubblesrt();

                System.out.println("Elements before sorting: ");
                bsrt.display(n, arr);

                bsrt.sort(n, arr);
                System.out.println("Elements after sorting: ");
                bsrt.display(n, arr);

                // TODO Auto-generated method stub

        }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Console
<terminated> Bubble_sort [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse.j
Enter the No of Elements to sort:
5
Enter 5 Elements
40
20
5
64
52
Elements before sorting:
40
20
5
64
52
Elements after sorting:
5
20
40
52
64
```

PROGRAM 26

AIM

Maintain a list of Strings using ArrayList from collection framework, perform built-in operations.

ALGORITHM

Step 1: Start

Step 2: Create an object ‘obj’ of type ArrayList.

Step 3: Put values into it using add()

Step 4: Manipulate the list using built in functions.

Step 5: Print the elements in a

Step 6: Stop

PROGRAM CODE

list_of_string.java	import java.util.ArrayList;
	import java.util.Collections;
	
	public class list_of_strings {
	
	public static void main(String[] args) {
	// TODO Auto-generated method stub
	ArrayList<String> obj = new ArrayList<String>();
	obj.add("Ford Mustang");
	obj.add("Lamborghini");
	obj.add("Ferrari");
	obj.add("Jeep Wrangler");
	obj.add("Dodge Challenger");
	
	//Displaying array list after add operation
	System.out.println("\n Display Array List : \n");
	for(String list:obj)
	System.out.println("\t"+list);
	
	//Removing list elements from array
	obj.remove("Ferrari");
	System.out.println("\n Display List after Removing : \n");
	for(String list:obj)
	System.out.println("\t"+list);

```
                    //Sorting Array list
                    System.out.println("\n Sorted Array list : \n");
                    Collections.sort(obj);
                    for(String list:obj)
                              System.out.println("\t"+list);

                    //Get element at an index value 2
                    System.out.println("\n Display Index element :
"+obj.get(2));

                    //Getting current size of list
                    System.out.println("\n Size of the Array list:
"+obj.size());

                    //Clearing the Array list
                    obj.clear();
                    System.out.println("\n Clear all elements in Array list
:"+obj);
          }

}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Console
<terminated> list_of_strings [Java Application] C:\Users\brahm\.p2\pool\plugins\org.eclipse

Display Array List :

        Ford Mustang
        Lamborghini
        Ferrari
        Jeep Wrangler
        Dodge Challenger

Display List after Removing :

        Ford Mustang
        Lamborghini
        Jeep Wrangler
        Dodge Challenger

Sorted Array list :

        Dodge Challenger
        Ford Mustang
        Jeep Wrangler
        Lamborghini

Display Index element : Jeep Wrangler

Size of the Array list: 4

Clear all elements in Array list :[]
```

PROGRAM 27

AIM

Program to remove all the elements from a linked list.

ALGORITHM

Step 1: Start

Step 2: Declare a 2 D array named str of type String and read values into it.

Step 3: Create an object student of type LinkedList and put values in str into stud using add()

Step 4: Traverse through stud using Iterator and print the values.

Step 5: Stop

PROGRAM CODE

co4qn11.java

```
package co4;

import java.util.LinkedList;
import java.util.Scanner;

public class co4qn11 {
        public static void main(String[] args) {
            int n;
            String data;
            LinkedList<String> ll = new LinkedList<String>();
            System.out.println("Enter the number of data");
            Scanner sc = new Scanner(System.in);
            n = sc.nextInt();
            System.out.println("Enter the data");
            sc.nextLine();
            for(int i=0;i<n;i++)
            {
              data = sc.nextLine();
              ll.add(data);
            }
            System.out.println("LinkedList: "+ll);
            System.out.println("All the elements removed from Linked
list");
            ll.clear();
            System.out.println(ll);
}
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Enter the number of data
6
Enter the data
Benz
BMW
Mustang
Toyata
Hyundai
Bugatti
LinkedList: [Benz, BMW, Mustang, Toyata, Hyundai, Bugatti]
All the elements removed from Linked list
[]
```

PROGRAM 28

AIM

Program to remove an object from the Stack when the position is passed as parameter.

ALGORITHM

Step 1: Start

Step 2: Create an object named ‘s’ of type Stack

Step 3: Read elements into fruits using add()

Step 4: Remove some elements using remove()

Step 5: Print the final stack

Step 6: Stop

PROGRAM CODE

co4qn12.java	

```
package co4;
import java.util.Scanner;
import java.util.Stack;

public class co4qn12 {
        public static void main(String[] args) {
            int n;
            String str;
            Stack<String> s = new Stack<String>();
            System.out.println("Enter the number of elements:");
            Scanner sc = new Scanner(System.in);
            n = sc.nextInt();
            sc.nextLine();
            System.out.println("Enter the elements:");
            for(int i=0;i<n;i++)
            {
               str = sc.nextLine();
               s.add(str);
            }
            System.out.println("\nStack elements:"+s);
            System.out.println("\nTop element:"+s.peek());
            System.out.println("Popped element:"+s.pop());
            System.out.println("Stack elements after popped:"+s);
```

```
            System.out.println("\nRemove Element at position 1:"+s.remove(0));
            System.out.println("Stack elements after removed:"+s);
            System.out.println("\nRemove Abraham Lincoln:");
            s.remove("Abraham Lincoln");
            System.out.println("Stack elements after removing Abraham Lincoln:"+s);
        }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Enter the number of elements:
7
Enter the elements:
George Bush
Donald Trump
Barack Obama
Abraham Lincoln
Bill Clinton
Ronald Reagan
Joe Biden

Stack elements:[George Bush, Donald Trump, Barack Obama, Abraham Lincoln, Bill Clinton, Ronald Reagan, Joe Biden]

Top element:Joe Biden
Popped element:Joe Biden
Stack elements after popped:[George Bush, Donald Trump, Barack Obama, Abraham Lincoln, Bill Clinton, Ronald Reagan]

Remove Element at position 1:George Bush
Stack elements after removed:[Donald Trump, Barack Obama, Abraham Lincoln, Bill Clinton, Ronald Reagan]

Remove Abraham Lincoln:
Stack elements after removing Abraham Lincoln:[Donald Trump, Barack Obama, Bill Clinton, Ronald Reagan]
```

PROGRAM 29

AIM

Program to demonstrate the creation of queue object using the PriorityQueue class.

ALGORITHM

Step 1: Start

Step 2: Create an object ‘stud’ of type PriorityQueue.

Step 3: Enter elements into stud using add()

Step 4: Remove some elements from stud using remove()

Step 5: Print the details with the help of Iterator

Step 6: Stop

PROGRAM CODE

co4qn13.java

```
package co4;
import java.util.Iterator;
import java.util.PriorityQueue;
import java.util.Scanner;

public class co4qn13 {
	public static void main(String[] args) {
		PriorityQueue<String> pq=new PriorityQueue<String>();
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter Number Of elements ");
		int n=sc.nextInt();
		System.out.println("Enter the elements ");
		for(int i =0;i<n;i++)
		{
			String st=sc.next();
			pq.add(st);

		}
		System.out.println("Iterating the queue elements\n ");
		Iterator<String> itr=pq.iterator();
		while(itr.hasNext()){
			System.out.println(itr.next());
		}
		pq.remove();
```

```
                pq.poll();
                System.out.println("After removing two elements \n");
                Iterator<String> itr2=pq.iterator();
                while(itr2.hasNext()){
                        System.out.println(itr2.next());
                }
        }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Enter Number Of elements
5
Enter the elements
Bently
Rolls-Royce
Bugatti
Porshe

Ferrari
Iterating the queue elements

Bently
Ferrari
Bugatti
Rolls-Royce
Porshe
After removing two elements

Ferrari
Rolls-Royce
Porshe
```

PROGRAM 30

AIM

Program to demonstrate the addition and deletion of elements in deque.

ALGORITHM

Step 1: Start

Step 2: Create a deque type object named ‘dq’.

Step 3: Put data into the dq using appropriate functions.

Step 4: Remove the data using built in functions.

Step 5: Print the data in dq

Step 6: Stop

PROGRAM CODE

co4qn14.java

```
package co4;
import java.util.Deque;
import java.util.LinkedList;
import java.util.Scanner;

public class co4qn14 {
        public static void main(String[] args) {
    int ch;
    String data;
    Deque<String> dq = new LinkedList<String>();
    Scanner sc = new Scanner(System.in);
    do
    {
        System.out.println("1.Insert the element at first");
        System.out.println("2.Insert the element at last");
        System.out.println("3.Delete the element at first");
        System.out.println("4.Delete the element at last");
        System.out.println("5.Display");
        System.out.println("6.Exit");
        System.out.println("\nEnter your choice:");
        ch = sc.nextInt();
        sc.nextLine();
        switch(ch)
        {
```

```
                case 1: System.out.println("Enter the element to be inserted at
first: ");
                        data = sc.nextLine();
                        dq.addFirst(data);
                        break;
                case 2: System.out.println("Enter the element to be inserted at
last: ");
                        data = sc.nextLine();
                        dq.addLast(data);
                        break;
                case 3: System.out.println("Element deleted from the first
position: ");
                        dq.removeFirst();
                        break;
                case 4: System.out.println("Element deleted from the last
position: ");
                        dq.removeLast();
                        break;
                case 5: System.out.println("Elements in list are: ");
                        System.out.println(dq);
                        break;
                case 6: System.exit(0);
                        break;
                default:System.out.println("Invalid Choice...Please enter a valid
choice!!!");
            }
        }while(true);
    }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter your choice:
1
Enter the element to be inserted at first:
10
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter your choice:
1
Enter the element to be inserted at first:
8
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter your choice:
1
Enter the element to be inserted at first:
7
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit
```

```
Enter your choice:
1
Enter the element to be inserted at first:
5
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter your choice:
1
Enter the element to be inserted at first:
2
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter your choice:
5
Elements in list are:
[2, 5, 7, 8, 10]
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter your choice:
2
Enter the element to be inserted at last:
25
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
```

```
5.Display
6.Exit

Enter your choice:
5
Elements in list are:
[2, 5, 7, 8, 10, 25]
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter your choice:
7
Invalid Choice...Please enter a valid choice!!!
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter your choice:
3
Element deleted from the first position:
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
5.Display
6.Exit

Enter your choice:
5
Elements in list are:
[5, 7, 8, 10, 25]
1.Insert the element at first
2.Insert the element at last
3.Delete the element at first
4.Delete the element at last
```

PROGRAM 31

AIM

Program to demonstrate the creation of Set object using the LinkedHashset class.

ALGORITHM

Step 1: Start

Step 2: Create a LinkedHashSet named co4qn15 and create an object named ‘s’ for the same.

Step 3: Add the elements into the object ‘s’ of type co4qn15 LinkedHashSet

Step 4: Getting the element to be deleted in LinkedHashSet from user and remove the element from LinkedHashSet

Step 5: After removal print the elements of LinkedHashSet

Step 6: Stop

PROGRAM CODE

co4qn15.java

LinkedHashset class

```
package co4;
import java.util.LinkedHashSet;
import java.util.Scanner;

public class co4qn15 {
	public static void main(String[] args) {

		LinkedHashSet<String> s = new LinkedHashSet<String>();
		int n;
		String x;
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter no of elements to be added: ");
		n=sc.nextInt();
		sc.nextLine();
		System.out.println("Enter set elements: ");
		for(int i=0;i<n;i++)
		{
			x=sc.nextLine();
			s.add(x);
		}
		System.out.println("Displaying LinkedHashSet:"+s);
```

```
                System.out.println("Size of LinkedHashSet: "+s.size());
                System.out.println("Enter element to be deleted:");
                String d=sc.nextLine();

                if(s.remove(d))
                {
                        System.out.println("Set after removal:"+s);
                }
                else
                {
                        System.out.println("Element not found!!");
                }
        }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Enter no of elements to be added:
5
Enter set elements:
Ferrari
Bugatti
Lamborghini
Lexus
Jaguar
Displaying LinkedHashSet:[Ferrari, Bugatti, Lamborghini, Lexus, Jaguar]
Size of LinkedHashSet: 5
Enter element to be deleted:
Lamborghini
Set after removal:[Ferrari, Bugatti, Lexus, Jaguar]
```

PROGRAM 32

AIM

Write a Java program to compare two hash set.

ALGORITHM

Step 1: Start

Step 2: Create an object named ‘set1’ of type HashSet.

Step 3: Add values into hashset using add() function.

Step 4: Create another object named ‘set2’ of type HashSet

Step 5: Add values into hashset using add() function.

Step 6: Create another object named ‘result_set’ of type HashSet

Step 7: While traversing through the hashset using for loop, compare the two hashset objects set1 and set2 using contain() function and print the same.

Step 8: Stop

PROGRAM CODE

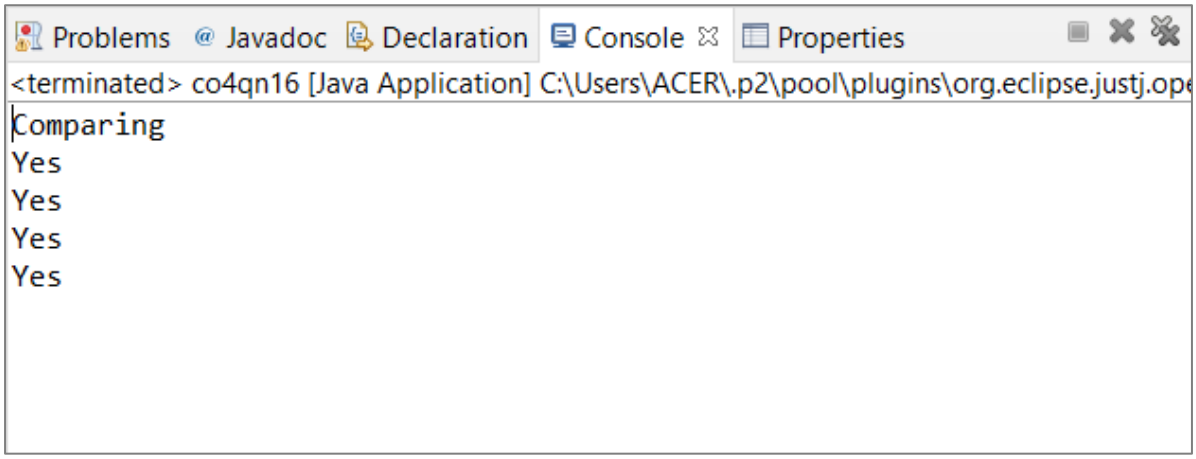
co4qn16.java

```
package co4;
import java.util.HashSet;

public class co4qn16
{
        public static void main(String[] args)
        {
    // Create a empty hash set
    HashSet<String> set1 = new HashSet<String>();

    // use add() method to add values in the hash set
    set1.add("Benz");
    set1.add("BMW");
    set1.add("Lamborghini");
    set1.add("Ferarri");

    HashSet<String> set2 = new HashSet<String>();
    set2.add("Benz");
```

```
        set2.add("BMW");
        set2.add("Lamborghini");
        set2.add("Ferarri");

        //comparison output in hash set
        System.out.println("Comparing");
        HashSet<String>result_set = new HashSet<String>();
        for (String element : set1){
            System.out.println( set2.contains(element) ? "Yes" : "No");
        }
    }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT



PROGRAM 33

AIM

Program to demonstrate the working of Map interface by adding, changing and removing elements.

ALGORITHM

Step 1: Start

Step 2: Create an object of type Map named ‘mp’

Step 3: Put values into mp using put() function and remove() function to remove the values

Step 4: print the final map after all operations

Step 5: Stop

PROGRAM CODE

co4qn17.java	package co4;
	import java.util.*;
	
	public class co4qn17 {
	
	public static void main(String args[])
	{
	
	Map<Integer, String> mp = new HashMap<>();
	//Inserting elements..
	System.out.println("Enter the limit:");
	Scanner inp = new Scanner(System.in);
	int n= inp.nextInt();
	System.out.println("Enter the Roll number and Name");
	while(n!=0) {
	
	int e= inp.nextInt();
	String s= inp.next();
	mp.put(e, s);
	n--;
	}
	
	System.out.println("Initial Map:"+mp);

```
                System.out.println("enter the num and name to update:");
                Scanner in = new Scanner(System.in);
                int e= in.nextInt();
                String s= in.next();
                mp.put(e, s);

                System.out.println("Updated Map:"+mp);

                //Removing..
                System.out.println("Enter the Roll number to be removed:");
                int r=inp.nextInt();
                mp.remove(r);

                // Final Map..

                System.out.println("After Removing the entry, Final Map
is:"+mp);

            }
        }
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> co4qn17 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.ope
Enter the limit:
3
Enter the Roll number and Name
101 Abhi
105 Devan
111 Rahul
Initial Map:{101=Abhi, 105=Devan, 111=Rahul}
enter the num and name to update:
105 Dhyan
Updated Map:{101=Abhi, 105=Dhyan, 111=Rahul}
Enter the Roll number to be removed:
111
After Removing the entry, Final Map  is:{101=Abhi, 105=Dhyan}
```

PROGRAM 34

AIM

Program to Convert HashMap to TreeMap.

ALGORITHM

Step 1: Start

Step 2: Create an object of type Map named ‘map’

Step 3: Add values into map object using put().

Step 4: To convert the Map type into TreeMap type, create an object of treeMap type and move all the values of map object using putAll() function.

Step 5: Print the values.

Step 6: Stop

PROGRAM CODE

co4qn18.java

```
package co4;

import java.util.*;

public class co4qn18 {

  public static void main(String args[]) {

            Map<String, String> map = new HashMap<>();
            System.out.println("Enter the limit:");
      Scanner sc = new Scanner(System.in);
      int n= sc.nextInt();
      System.out.println("Enter the Roll number and Name");
      while(n!=0) {

          String e= sc.next();
          String s= sc.next();
          map.put(e, s);
          n--;
      }

     System.out.println("HashMap:"+map);
```

```
        Map<String, String> treeMap = new TreeMap<>();
        treeMap.putAll(map);
        System.out.println("TreeMap:"+treeMap);
    }
}
```

RESULT: The above program is successfully executed and obtained the output

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> co4qn18 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.op
Enter the limit:
3
Enter the Roll number and Name
114 Hareesh
120 Jayadev
136 Madhav
HashMap:{114=Hareesh, 136=Madhav, 120=Jayadev}
TreeMap:{114=Hareesh, 120=Jayadev, 136=Madhav}
```

PROGRAM 35

AIM

Program to draw Circle, Rectangle, Line in Applet

ALGORITHM

Step.1: Start

Step.2: Define a class ‘q2’ that extends Applet class.

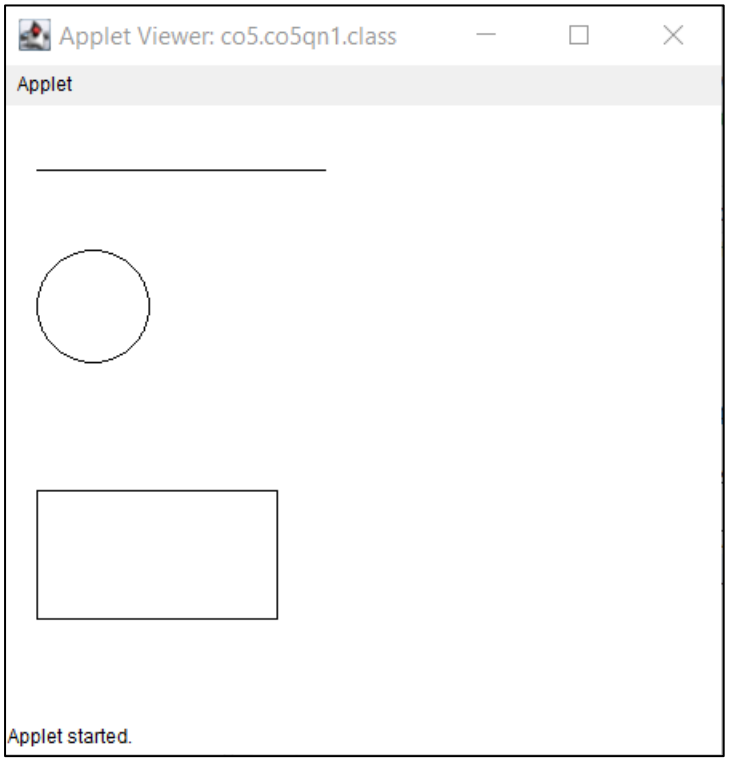
Step.3: Draw a line, rectangle and circle using drawLine, drawRect and drawOval methods of Graphics class respectively.

Step.4: Stop

PROGRAM CODE

q1.java	import java.applet.Applet;
	import java.awt.*;
	
	public class q1 extends Applet {
	  public void paint(Graphics g) {
	  g.drawLine(20, 20, 200, 20);
	  g.drawRect(20, 60, 200, 40);
	  g.drawOval(20, 120, 200, 160);
	  }
	
	}

OUTPUT



PROGRAM 36

AIM

Program to find maximum of three numbers using AWT.

ALGORITHM

Step.1: Start the program.

Step.2: Define a class ‘q2’ that extends Applet class and implements ActionListener interface.

Step.3: Using TextField class object, construct the required no. of textfields wide enough to hold the values entered by the user.

Step.4: Using Button class object, construct a labeled button that sends an instance of ActionEvent.

Step.5: Call addActionListener() method to send events from the button to the new listener.

Step.6: Get the string values from textfields and then parse them as integers.

Step.7: Compare each value using if-else statements to find the maximum value and set the result accordingly.

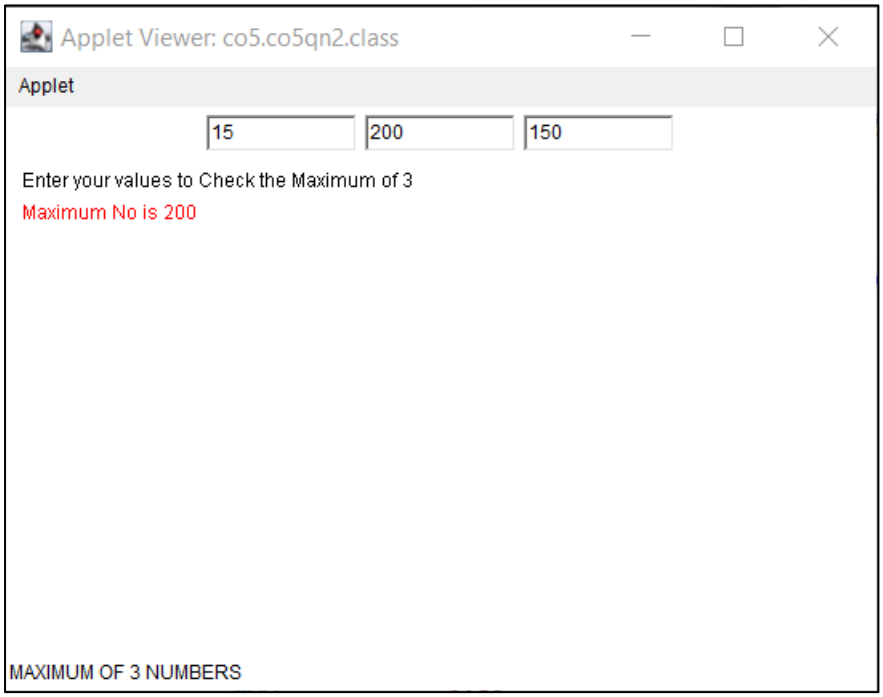
Step.8: Stop the program.

PROGRAM CODE

q2.java	import java.applet.*;
	import java.awt.*;
	public class q2 extends Applet {
	  TextField T1,T2,T3;
	
	  public void init(){
	  T1 = new TextField(10);
	  T2 = new TextField(10);
	  T3 = new TextField(10);

```
        add(T1);
        add(T2);
        add(T3);

        T1.setText("0");
        T2.setText("0");
        T3.setText("0");
    }

    public void paint(Graphics g){
        int a, b, c,result;
        String str;

        g.drawString("Enter value to Check the Maximum of 3 ",10,50);

        str=T1.getText();
        a=Integer.parseInt(str);
        str=T2.getText();
        b=Integer.parseInt(str);
        str=T3.getText();
        c=Integer.parseInt(str);

        g.setColor(Color.blue);
        if (a>b) {
            if (a>c)
                result=a;
            else
                result=c;
        }
        else{
            if (b>c)
                result=b;
            else
                result=c;
        }
        g.drawString("Maximum of 3 No is "+result,10,70);
        showStatus("MAXIMUM OF 3 NUMBERS");
    }

    public boolean action(Event e, Object o){
        repaint();
        return true;
    }
}
```

OUTPUT


Applet Viewer: co5.co5qn2.class
Applet
15
200
150
Enter your values to Check the Maximum of 3
Maximum No is 200
MAXIMUM OF 3 NUMBERS

PROGRAM 37

AIM

Find the percentage of marks obtained by a student in 5 subjects. Display a happy face if he secures above 50% or a sad face if otherwise.

ALGORITHM

Step.1: Start the program.

Step.2: Define a class ‘q3’ that extends Applet class and implements ActionListener interface.

Step.3: Using TextField class object, construct textfields to receive marks of 5 subjects from the user.

Step.4: Using Button class object, construct a labeled button that sends an instance of ActionEvent.

Step.5: Call addActionListener() method to send events from the button to the new listener.

Step.6: Get the string values from textfields and then parse them as float values.

Step.7: Calculate the percentage:

Percent = ((mark1+mark2+mark3+mark4+mark5)*100)/500

Step.8: Define a paint() method that contains functions from Graphics class to display a happy face if student secures above 50% or a sad face if otherwise

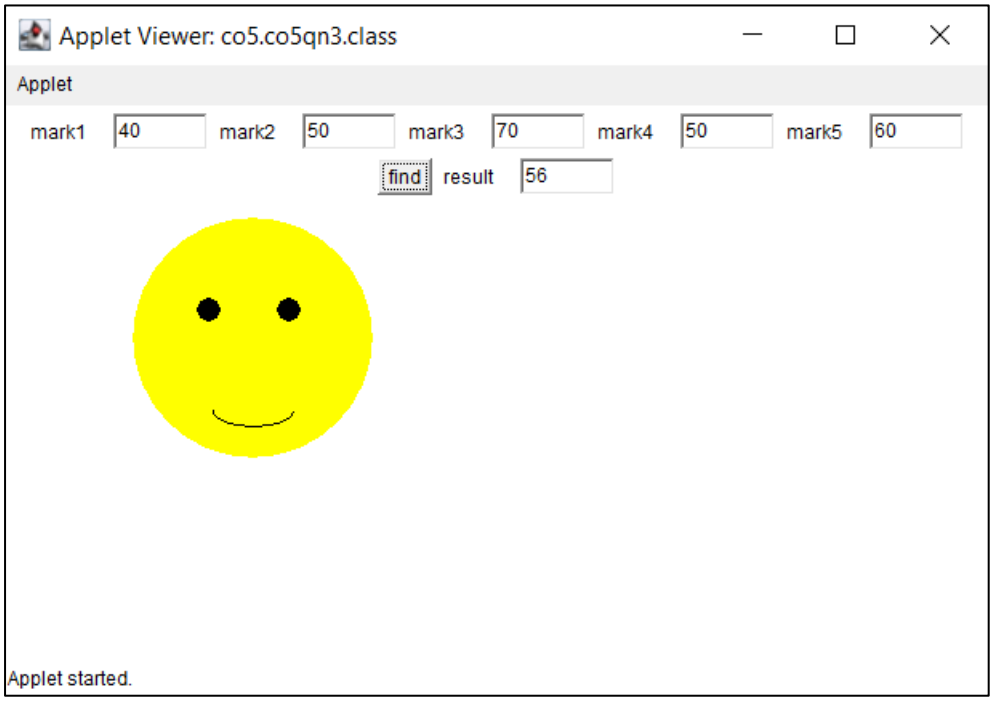
Step.9: Stop the program.

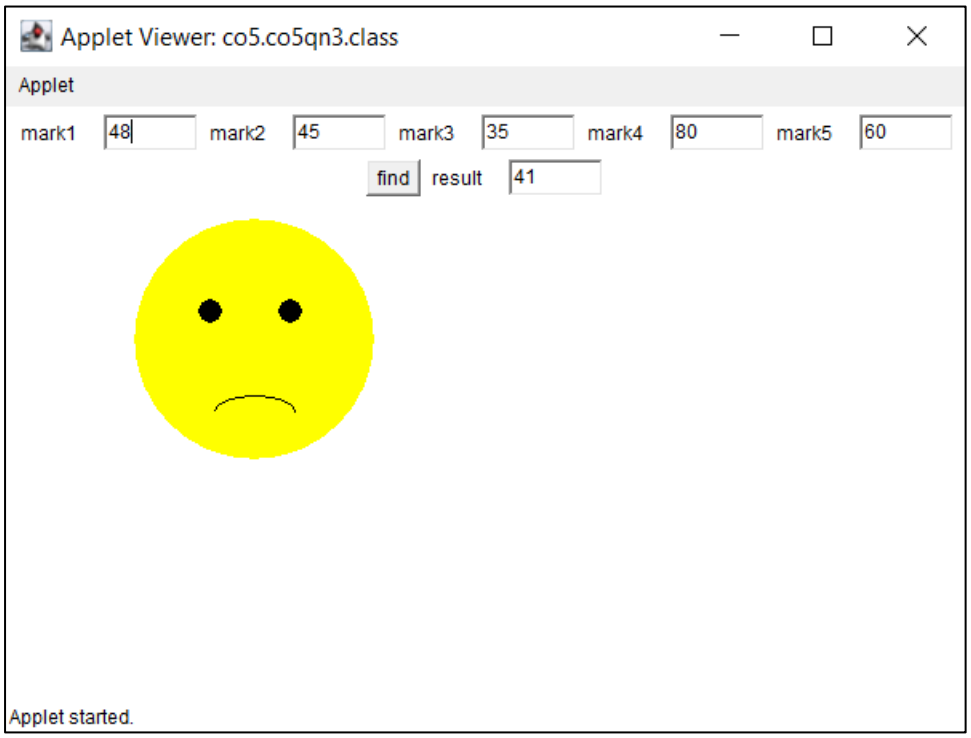
PROGRAM CODE :

q3.java	import java.applet.Applet;
	import java.awt.*;
	import java.util.*;
	
	public class q3 extends Applet {
	
	public void paint(Graphics g){
	
	System.out.println("Enter marks : ");
	int marks = 0;
	int n =3;

```
        Scanner sc = new Scanner(System.in);
        while(n!=0)
        {
            marks = marks + sc.nextInt();
            n--;
        }

        int per;
        per = marks /3;

        System.out.println("Percentage is :" + per);



        if(per > 50)
        {
            // Oval for face outline
            g.drawOval(80, 70, 150, 150);

            // Ovals for eyes
            // with black color filled
            g.setColor(Color.BLACK);
            g.fillOval(120, 120, 15, 15);
            g.fillOval(170, 120, 15, 15);

            // Arc for the smile
            g.drawArc(130, 180, 50, 20, 180, 180);
        }
        else
        {
            // Oval for face outline
            g.drawOval(80, 70, 150, 150);

            // Ovals for eyes
            // with black color filled
            g.setColor(Color.BLACK);
            g.fillOval(120, 120, 15, 15);
            g.fillOval(170, 120, 15, 15);

            // Arc for the smile
            g.drawArc(130, 180, 100, 20, 90, 45);
        }

    }


}
```

OUTPUT


Applet Viewer: co5.co5qn3.class
Applet
mark1
40
mark2
50
mark3
70
mark4
50
mark5
60
find
result
56
Applet started.


Applet Viewer: co5.co5qn3.class
Applet
mark1
48
mark2
45
mark3
35
mark4
80
mark5
60
find
result
41
Applet started.

PROGRAM 38

AIM

Using 2D graphics commands in an Applet, construct a house. On mouse click event, change the color of the door from blue to red.

ALGORITHM

Step.1: Start the program.

Step.2: Define a class ‘house’ that extends Applet and implements MouseListener.

Step.3: Define methods to add MouseListener to the panel.

Step.4: Using getX() and getY() methods, get the coordinates of the door to repaint when the MousePressed event occurs.

Step.5: Stop the program.

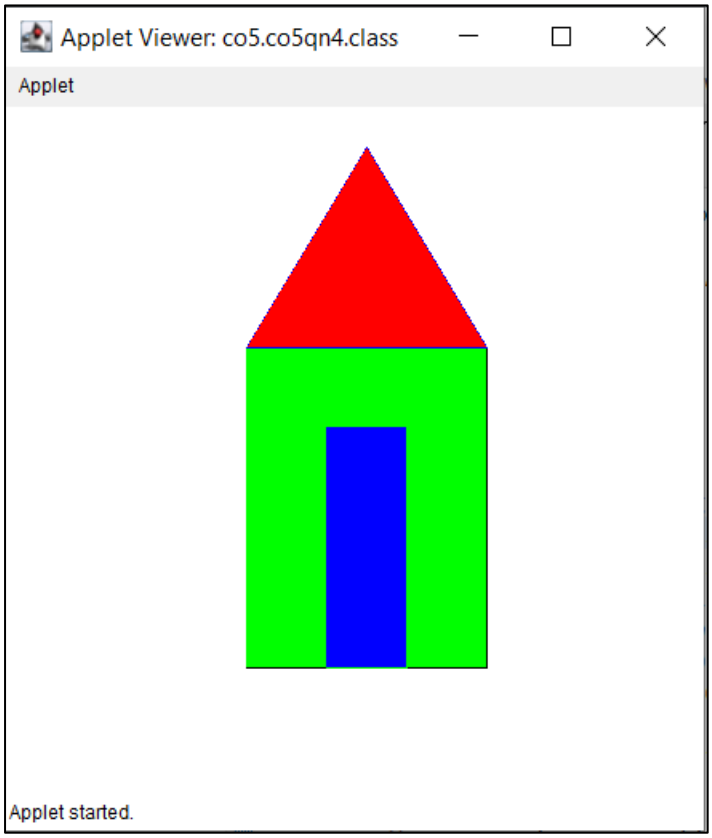
PROGRAM CODE :

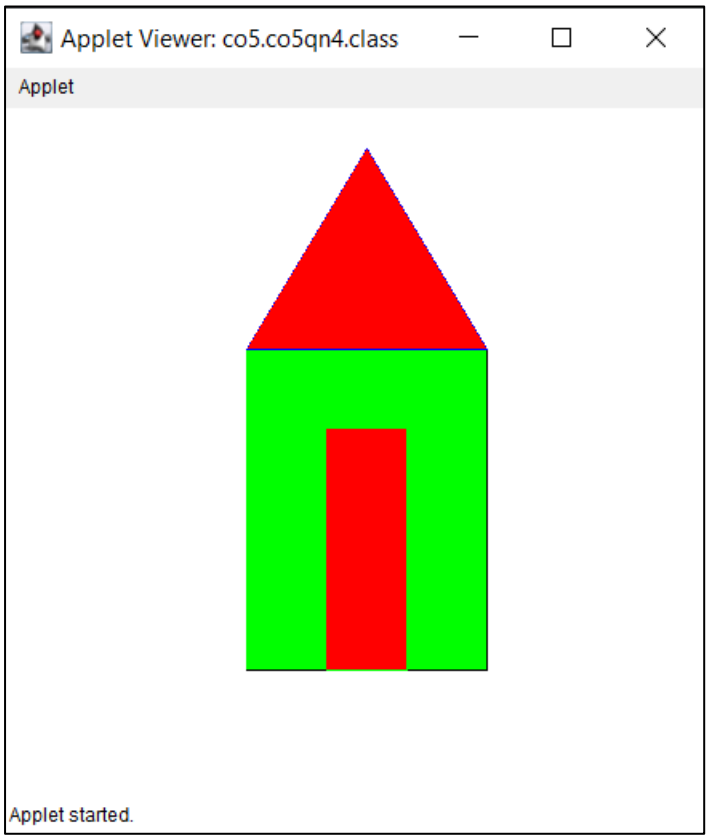
q4.java

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

/*<applet code = "house.class" width = "600" height = "600"></applet>*/

public class q4 extends Applet implements MouseListener
{
   int a,b;
   public void init()
   {
      addMouseListener(this);
   }
   public void paint(Graphics g){
      g.drawRect(10,50,50,100); //house
      g.setColor(Color.green);
      g.fillRect(10,50,50,100);
      g.drawRect(25,100,20,50); //door
      g.setColor(Color.white);
      g.fillRect(25,100,20,50);
      g.drawArc(10,30,50,40,0,180);
      g.setColor(Color.red);
```

```
        g.fillArc(10,30,50,40,0,180);

        if(a>25 && a<100 && b>25 && b<100)
        {
            g.setColor(Color.red);
            g.fillRect(25,100,20,50);
        }
    }
    public void mouseClicked(MouseEvent e){

    }
    public void mouseEntered(MouseEvent e){

    }
    public void mouseExited(MouseEvent e) {

    }
    public void mousePressed(MouseEvent e){

        a=e.getX();
        b=e.getY();
        repaint();
    }
    public void mouseReleased(MouseEvent e){

    }
}
```

OUTPUT


Applet Viewer: co5.co5qn4.class
Applet
Applet started.


Applet Viewer: co5.co5qn4.class
Applet
Applet started.

PROGRAM 39

AIM

Implement a simple calculator using AWT components

ALGORITHM

Step 1: Start

Step 2: Define a class ‘q5’ that extends implements ActionListener interface.

Step 3: Using TextField class object, construct the required no. of textfields wide enough to hold the values entered by the user.

Step 4: Using Label class object, construct and provide the appropriate labels.

Step 5: Using Button class object, construct labeled buttons that send the instances of ActionEvent.

Step 6: Call addActionListener() method to send events from the button to the new listener.

Step 7: Get the string values from textfields and then parse them as integers.

Step 8: Perform various methods to add, subtract, multiply and divide those integers.

Step 9: Stop

PROGRAM CODE :

q5.java

```
import java.awt.*;
import java.awt.event.*;
public class q5 implements ActionListener
{
   int c,n;
   String s1,s2,s3,s4,s5;
   Frame f;
   Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;
   Panel p;
   TextField tf;
   GridLayout g;
   q5()
```

```
    {
        f = new Frame("My calculator");
        p = new Panel();
        f.setLayout(new FlowLayout());
        b1 = new Button("0");
        b1.addActionListener(this);
        b2 = new Button("1");
        b2.addActionListener(this);
        b3 = new Button("2");
        b3.addActionListener(this);
        b4 = new Button("3");
        b4.addActionListener(this);
        b5 = new Button("4");
        b5.addActionListener(this);
        b6 = new Button("5");
        b6.addActionListener(this);
        b7 = new Button("6");
        b7.addActionListener(this);
        b8 = new Button("7");
        b8.addActionListener(this);
        b9 = new Button("8");
        b9.addActionListener(this);
        b10 = new Button("9");
        b10.addActionListener(this);
        b11 = new Button("+");
        b11.addActionListener(this);
        b12 = new Button("-");
        b12.addActionListener(this);
        b13 = new Button("*");
        b13.addActionListener(this);
        b14 = new Button("/");
        b14.addActionListener(this);
        b15 = new Button("%");
        b15.addActionListener(this);
        b16 = new Button("=");
        b16.addActionListener(this);
        b17 = new Button("C");
        b17.addActionListener(this);
        tf = new TextField(20);
        f.add(tf);
        g = new GridLayout(4,4,10,20);
        p.setLayout(g);

p.add(b1);p.add(b2);p.add(b3);p.add(b4);p.add(b5);p.add(b6);p.add(b7);p.add(b8);
p.add(b9);

p.add(b10);p.add(b11);p.add(b12);p.add(b13);p.add(b14);p.add(b15);p.add(b16);p
.add(b17);
        f.add(p);
        f.setSize(300,300);
```

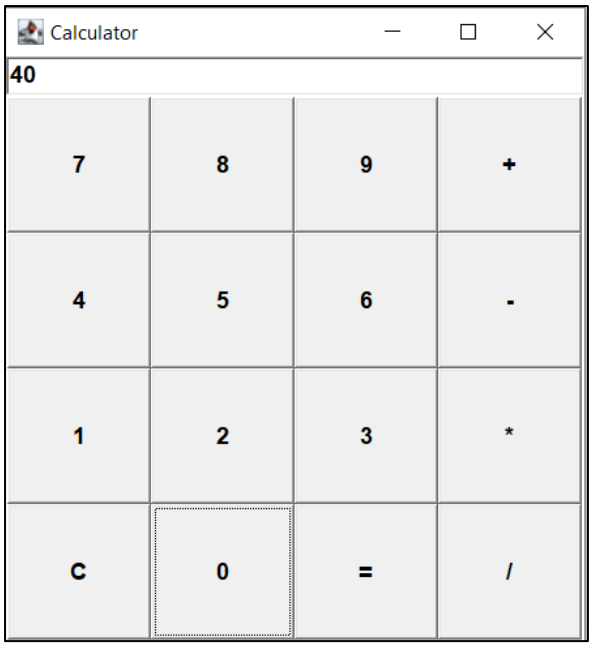
```
        f.setVisible(true);
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==b1)
        {
            s3 = tf.getText();
            s4 = "0";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b2)
        {
            s3 = tf.getText();
            s4 = "1";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b3)
        {
            s3 = tf.getText();
            s4 = "2";
            s5 = s3+s4;
            tf.setText(s5);
        }if(e.getSource()==b4)
    {
        s3 = tf.getText();
        s4 = "3";
        s5 = s3+s4;
        tf.setText(s5);
    }
        if(e.getSource()==b5)
        {
            s3 = tf.getText();
            s4 = "4";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b6)
        {
            s3 = tf.getText();
            s4 = "5";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b7)
        {
            s3 = tf.getText();
            s4 = "6";
            s5 = s3+s4;
```

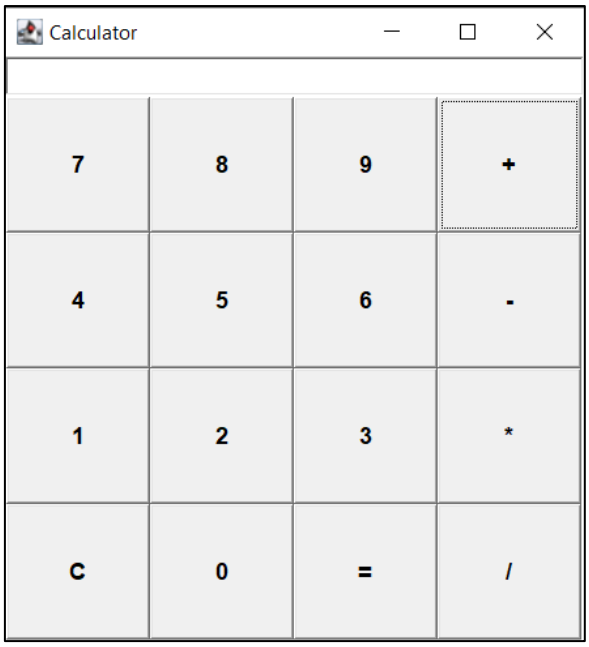
```
            tf.setText(s5);
        }
        if(e.getSource()==b8)
        {
            s3 = tf.getText();
            s4 = "7";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b9)
        {
            s3 = tf.getText();
            s4 = "8";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b10)
        {
            s3 = tf.getText();
            s4 = "9";
            s5 = s3+s4;
            tf.setText(s5);
        }
        if(e.getSource()==b11)
        {
            s1 = tf.getText();
            tf.setText("");
            c=1;

        }
        if(e.getSource()==b12)
        {
            s1 = tf.getText();
            tf.setText("");
            c=2;

        }
        if(e.getSource()==b13)
        {
            s1 = tf.getText();
            tf.setText("");
            c=3;

        }
        if(e.getSource()==b14)
        {
            s1 = tf.getText();
            tf.setText("");
            c=4;
```

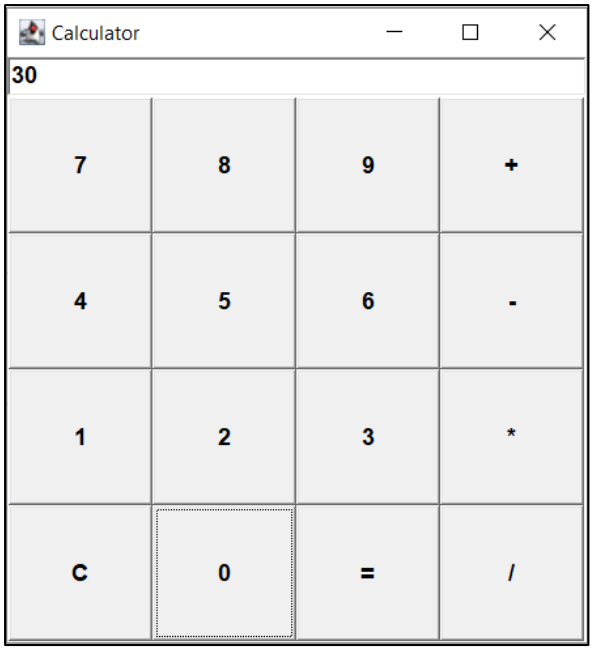
```
        }
        if(e.getSource()==b15)
        {
            s1 = tf.getText();
            tf.setText("");
            c=5;

        }
        if(e.getSource()==b16)
        {
            s2 = tf.getText();
            if(c==1)
            {
                n = Integer.parseInt(s1)+Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
            else
            if(c==2)
            {
                n = Integer.parseInt(s1)-Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
            else
            if(c==3)
            {
                n = Integer.parseInt(s1)*Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
            if(c==4)
            {
                try
                {
                    int p=Integer.parseInt(s2);
                    if(p!=0)
                    {
                        n = Integer.parseInt(s1)/Integer.parseInt(s2);
                        tf.setText(String.valueOf(n));
                    }
                    else
                        tf.setText("infinite");

                }
                catch(Exception i){}
            }
            if(c==5)
            {
                n = Integer.parseInt(s1)%Integer.parseInt(s2);
                tf.setText(String.valueOf(n));
            }
        }
```

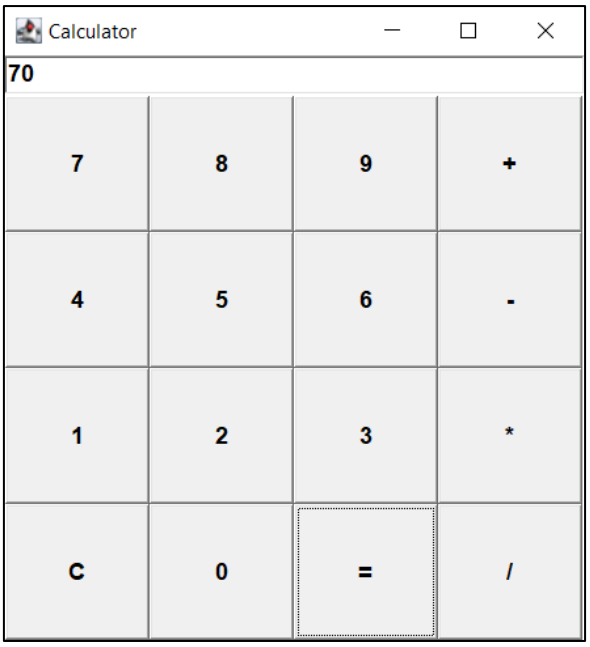
```
        if(e.getSource()==b17)
        {
            tf.setText("");
        }
    }

    public static void main(String[] abc)
    {
        q5 v = new q5();
    }
}
```

OUTPUT






Calculator
30
7
8
9
+
4
5
6
-
1
2
3
*
C
0
=
/


Calculator
70
7
8
9
+
4
5
6
-
1
2
3
*
C
0
=
/

PROGRAM 40

AIM

Develop a program that has a Choice component which contains the names of shapes such as rectangle, triangle, square and circle. Draw the corresponding shapes for given parameters as per user's choice

ALGORITHM

Step 1: Start

Step 2: Define a class 'shapes' that extends Applet class and implements ItemListener interface.

Step 3: Declare a new constructor of the Choice class to create an empty Choice menu.

Step 4: Use add() method to include items in the menu.

Step 5: Using getSelectedItem() method, get the item chosen by the user and repaint accordingly.

Step 6: Stop

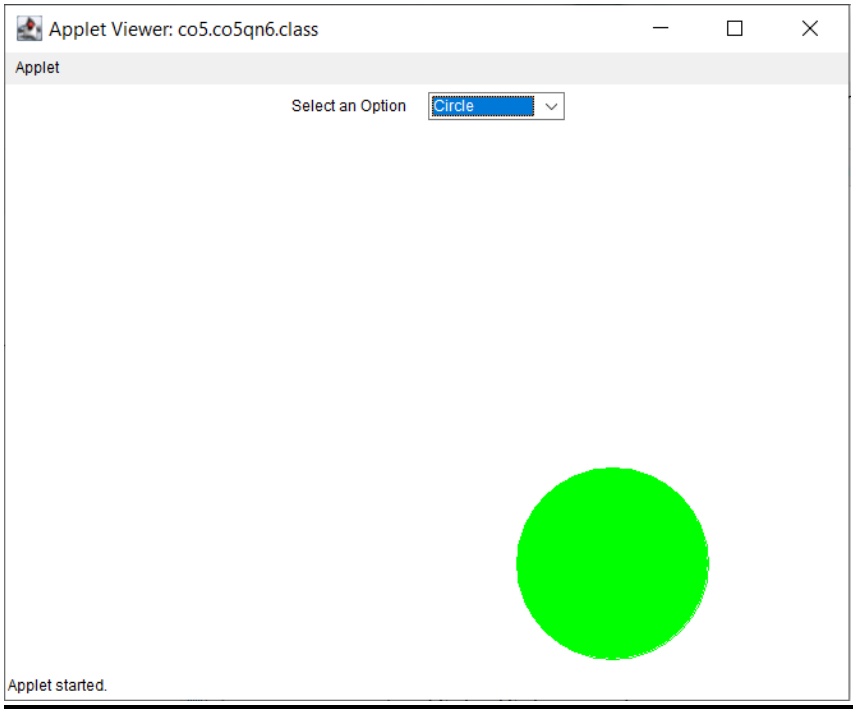
PROGRAM CODE

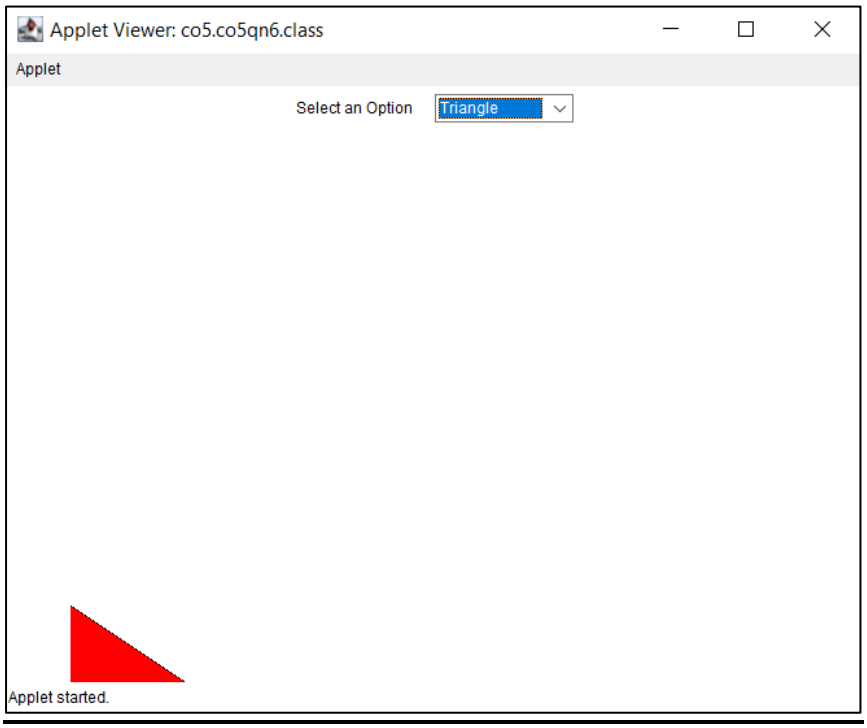
q6.java

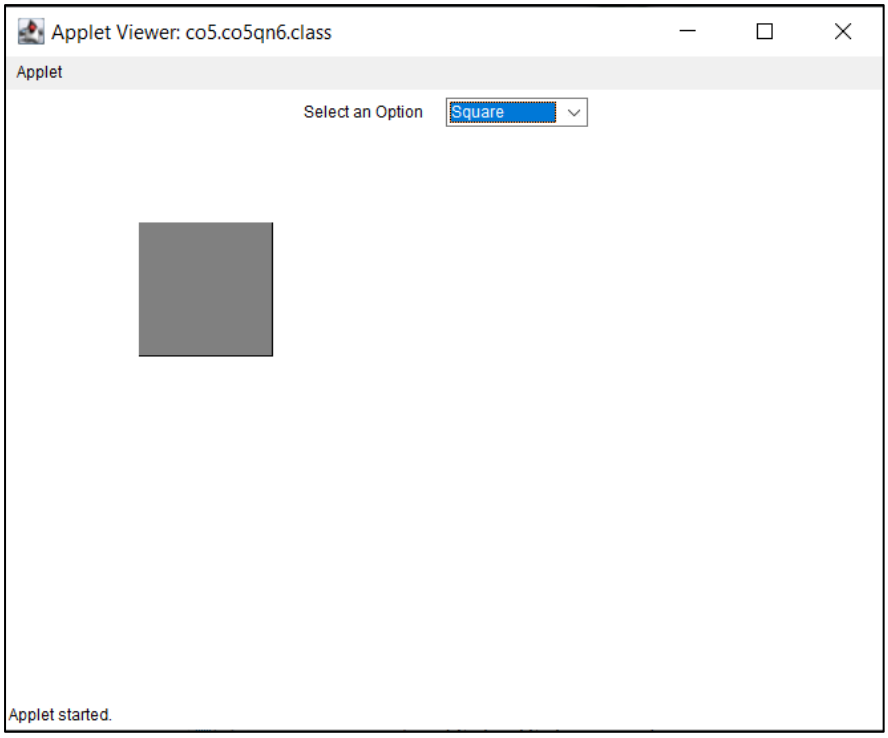
```
package shad;
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class draw_choice extends Applet implements ItemListener {

   Choice choice;
   int c;
   Label title;
   public void init()
   {
         title = new Label("SELECT A SHAPE:  ");
      choice = new Choice();
      choice.addItem("SHAPES");
      choice.addItem("RECTANGLE");
      choice.addItem("TRIANGLE");
      choice.addItem("SQUARE");
```

```
        choice.addItem("CIRCLE");
        add(title);
        add(choice);
        choice.addItemListener(this);
    }

  public void itemStateChanged (ItemEvent e)
    {
         c= choice.getSelectedIndex();
         repaint();
    }

  public void paint(Graphics g)
            {
         super.paint(g);

         if (c == 1)
         {
            g.drawString(choice.getItem(1),220,235);
            g.drawRect(150,70,200,150);
            g.fillRect(150,70,200,150);
         }
         if (c == 2)
         {
             g.drawString(choice.getItem(2),45,205);
             int[] x={80,160,5};
             int[] y={70,170,170};
             g.drawPolygon(x,y,3);
             g.fillPolygon(x,y,3);
         }
         if (c == 3)
         {
             g.drawString(choice.getItem(3),200,265);
             g.drawRect(200,200,50,50);
             g.fillRect(200,200,50,50);

         }
         if (c ==4)
         {
             g.drawString(choice.getItem(4),190,290);
             g.drawOval(170,170,90,90);
             g.fillOval(170,170,90,90);
         }
            }
}
/*
<applet code="draw_choice.class" width="500" height="700" border="2">
</applet>
*/
```

OUTPUT


Applet Viewer: co5.co5qn6.class
Applet
Select an Option
Circle
Applet started.


Applet Viewer: co5.co5qn6.class
Applet
Select an Option
Triangle
Applet started.


Applet Viewer: co5.co5qn6.class
Applet
Select an Option
Square
Applet started.

PROGRAM 41

AIM

Develop a program to handle all mouse events and window events

ALGORITHM

Step 1: Start

Step 2: Define a class mouseevents that extends Applet class and implements MouseListener interface.

Step 3: Define methods to add MouseListener to the panel which will have the following methods:

- void mouseClicked(MouseEvent me) - Invoked when the mouse has been clicked.
- void mousePressed(MouseEvent me) - Invoked when the mouse has been pressed.
- void mouseReleased(MouseEvent me) - Invoked when the mouse has been released.
- void mouseEntered(MouseEvent me) - Invoked when the mouse has entered the panel.
- void mouseExited(MouseEvent me) - Invoked when the mouse has exited the panel.
- void mouseDragged(MouseEvent me) - Invoked when the mouse has been dragged.

Step 4: Using getX() and getY() methods, get the location (or movements) of mouse pointer on the panel. Use them to display the necessary message in the output.

Step 5: Define another class WindowEvents that extends Applet class and implements WindowListener interface.

Step 6: Define methods to add WindowListener to the panel which will have the following

methods:

- void windowActivated(WindowEvent arg0) - Invoked when the window has been activated.
- void windowOpened(WindowEvent arg0) - Invoked when the window has been Opened.
- void windowDeactivated(WindowEvent arg0) - Invoked when the window has been deactivated.
- void windowIconified(WindowEvent arg0) - Invoked when the window has been iconified.
- void windowDeiconified(WindowEvent arg0) - Invoked when the window has been deiconified
- void windowClosed(WindowEvent arg0) - Invoked when the window has been closed.

Step 7: Display the appropriate message in the output.

Step 8: Stop

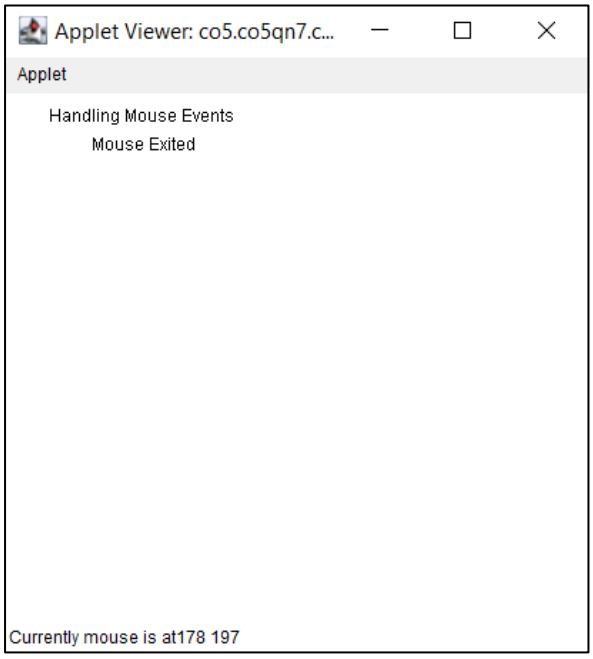
PROGRAM CODE :

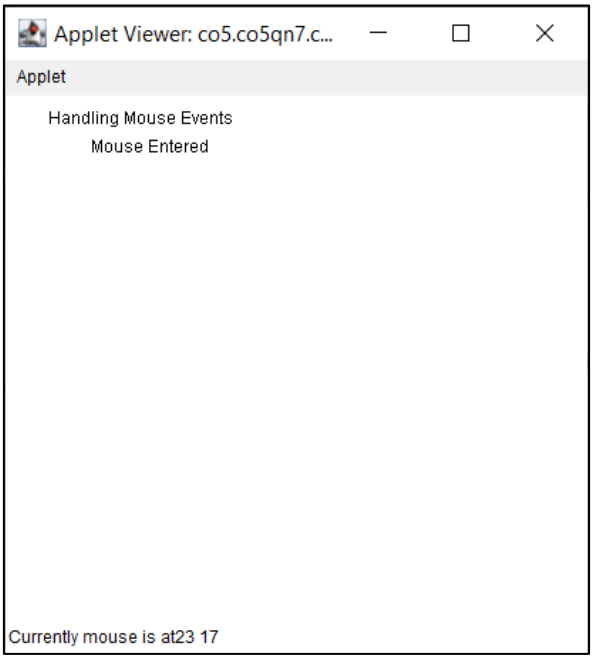
q7.java	package shad;
	import java.awt.*;
	import java.applet.*;
	import java.awt.event.*;
	import java.awt.event.WindowEvent;
	import java.awt.event.WindowListener;
	/*<applet code="mouse_events" width=300 height=300>
	</applet>*/
	public class mouse_events extends Applet implements
	MouseListener,MouseMotionListener
	{
	int x=0;
	int y=0;
	String msg="";
	
	public void init()
	{
	addMouseListener(this);

```
                addMouseMotionListener(this);
        }
        public void mouseClicked(MouseEvent me)
        {
                x=20;
                y=40;
                msg="Mouse Clicked";
                repaint();
        }
        public void mousePressed(MouseEvent me)
        {
                x=30;
                y=60;
                msg="Mouse Pressed";
                repaint();
        }
        public void mouseReleased(MouseEvent me)
        {
                x=30;
                y=60;
                msg="Mouse Released";
                repaint();
        }
        public void mouseEntered(MouseEvent me)
        {
                x=40;
                y=80;
                msg="Mouse Entered";
                repaint();
        }
        public void mouseExited(MouseEvent me)
        {
                x=40;
                y=80;
                msg="Mouse Exited";
                repaint();
        }
        public void mouseDragged(MouseEvent me)
        {
                x=me.getX();
                y=me.getY();
                showStatus("Currently mouse dragged"+x+" "+y);
                repaint();
        }
        public void mouseMoved(MouseEvent me)
        {
                x=me.getX();
                y=me.getY();
                showStatus("Currently mouse is at"+x+" "+y);
                repaint();
```

```
        }
        public void paint(Graphics g)
        {
                g.drawString("-----Handling Mouse Events-----",30,20);
                g.drawString(msg,60,40);
        }
}
```

OUTPUT





PROGRAM 42

AIM

Develop a program to handle Key events

ALGORITHM

Step.1: Start

Step.2: Define a class *keys* that extends Applet and implements KeyListener.

Step.3: Define methods to add KeyListener to the panel which will have the following methods:

- *void keyTyped(KeyEvent e)* – Invoked when a key has been typed.
- *void keyPressed(KeyEvent e)* - Invoked when a key has been pressed.
- *void keyReleased(KeyEvent e)* - Invoked when a key has been released.

Step.4: Using *getKeyChar()*, get the unicode and character representation of the

key pressed. Usethem to display the necessary message in the output.

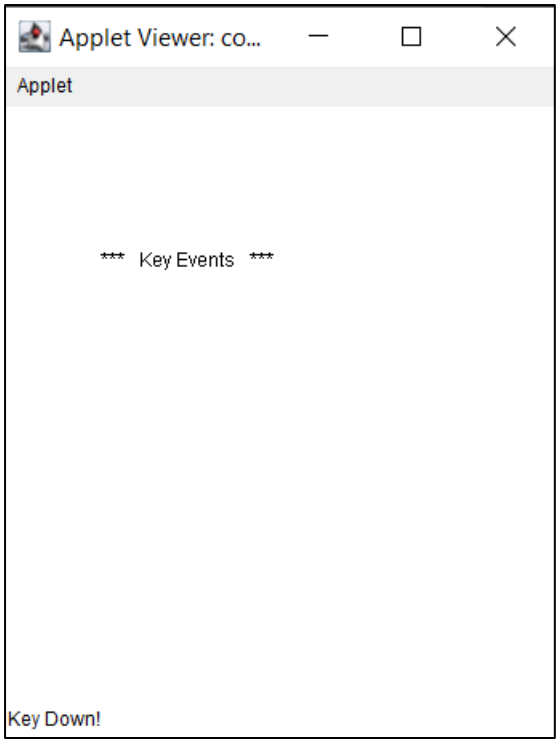
Step.5: Stop

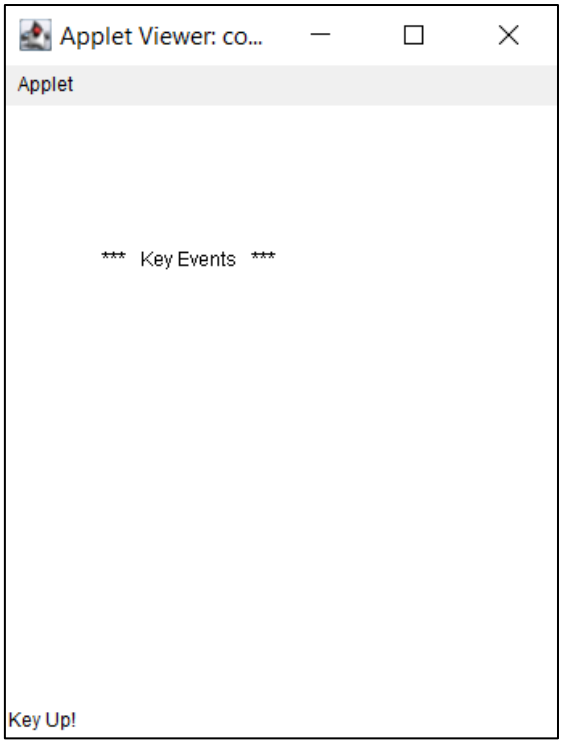
PROGRAM CODE

q8.java	package shad;
	import java.awt.*;
	import java.awt.event.*;
	/* <applet code="key_events.class" width=700 height=700></applet> */
	public class key_events extends Frame implements KeyListener
	{
	Label l;
	TextArea area;
	public key_events()
	{
	l=new Label();
	l.setBounds(20,50,200,40);
	area=new TextArea();
	area.setBounds(20,100,200,100);
	area.addKeyListener(this);
	
	add(l);

```
        add(area);
        setSize(400,400);
        setLayout(null);
        setVisible(true);
    }
    public void keyPressed(KeyEvent e)
    {
        l.setText("Key Pressed");
    }
    public void keyReleased(KeyEvent e)
    {
        l.setText("Key Released");
    }
    public void keyTyped(KeyEvent e)
    {
        l.setText("Key Typed");
    }
    public static void main(String[] args)
    {
        new key_events();
    }
}
```

OUTPUT





PROGRAM 43

AIM

Program to list the sub directories and files in a given directory and also search for a file name

ALGORITHM

Step.1: Start the program.

Step.2: Create a class named '*co6qn1*'

Step.3: Create an object for the *class File* to to initialize its constructor with the file source.

Step.4: Using list(), get the names of all the files present in the directory.

Step.5: Filter accordingly and store the file names to the list.

Step.6: Display the list.

Step.7: Stop the program.

PROGRAM CODE

co6qn1
.java
	package co6;
	
	import java.io.*;
	
	class co6qn1 {
	public static void main(String[] args)
	{
	//create a file object
	File directory = new File("E:\\MCA\\Semester 2\\Java Programming\\Programs\\co4\\src");
	
	String[] fileList = directory.list();
	int flag = 0;
	if(fileList == null)
	{
	System.out.println("Empty directory");
	}
	else
	{

```
                        for(String str : fileList)
                        {
                                System.out.println(str);
                        }
                }

                for(int i=0; i < fileList.length; i++)
                {
                        String filename  = fileList[i];
                        if(filename.equalsIgnoreCase("area.java"))
                        {
                                System.out.println("\n\n" +filename + "
found");
            flag = 1;
                        }
                }
                if(flag == 0)
                {
        System.out.println("\n\n File Not Found");
      }
        }
}
```

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> co6qn1 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.ope
Areaqn1.java
Arithmetic
Authenticate.java
AvgofPositive.java
Bubble_sort.java
co4
Fibonacci_Even.java
Graphics
Inputsqn2.java
list_of_strings.java
MultiThread.java
prod_cons.java
Stack.java


Stack.java found
```

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> co6qn1 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.ope
Areaqn1.java
Arithmetic
Authenticate.java
AvgofPositive.java
Bubble_sort.java
co4
Fibonacci_Even.java
Graphics
Inputsqn2.java
list_of_strings.java
MultiThread.java
prod_cons.java
Stack.java


File Not Found
```

PROGRAM 44

AIM

Write a program to write to a file, then read from the file and display the contents on the console.

ALGORITHM

Step.1: Start

Step.2: Create a class named ‘co6qn2’.

Step.3: Create an object of the *class File* to initialize its constructor with the file source.

Step.4: Create and use an object for the *FileWriter class* to write the file.

Step.5: Create and use an object for the *BufferedReader class* to read the stream of characters thespecified file.

Step.6: Display the contents read from the file on the console.

Step.7: Stop

PROGRAM CODE

co6qn2.java

```
package co6;

import java.io.FileWriter;
import java.io.FileReader;
import java.io.BufferedReader;
import java.io.IOException;

public class co6qn2 {
	public static void main(String[] args)
	{
		try
		{
			FileWriter fw = new FileWriter("file1.txt");
			fw.write("Hi User welcome to Java Programming
\n");
			fw.write("Here is your first program");
			fw.close();

			FileReader fr = new FileReader("file1.txt");
```

```
                    BufferedReader br = new BufferedReader(fr);
                    System.out.println("file content: ");
        String line;
        while((line = br.readLine()) != null)
        {
            System.out.println(line);
        }
                    fr.close();
                }
                catch (IOException e)
                {
                    System.out.println("An error occurred!!!....");
                }
        }
}
```

OUTPUT

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> co6qn2 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.ope
file content:
Hi User welcome to Java Programming
Here is your first program
```

PROGRAM 45

AIM

Write a program to copy one file to another.

ALGORITHM

Step.1: Start

Step.2: Create a class named '*co6_q3*'.

Step.3: Create and use an object for the *BufferedReader class* to read the stream of characters fromthe specified file.

Step.4: Create and use an object for the *FileWriter class* to write the stream of characters read bythe *BufferedReader*, to the file.

while ((s = br.readLine()) != null) {

fw.write(s);

}

Step.6: Display the appropriate message on the console.

Step.7: Stop.

PROGRAM CODE

co6qn3.java	package co6; import java.io.File; import java.io.FileInputStream; import java.io.FileOutputStream; import java.io.IOException; public class co6qn3 { public static void main(String[] args) { FileInputStream instream = null; FileOutputStream outstream = null;

```
                    try {
                                File infile = new
File("C:\\Users\\ACER\\OneDrive\\Desktop\\Java Programs\\Cristiano.txt");
                                File outfile = new
File("C:\\Users\\ACER\\OneDrive\\Desktop\\Java
Programs\\Player_details.txt");

                                instream = new FileInputStream(infile);
                                outstream = new FileOutputStream(outfile);

                                byte[] buffer = new byte[1024];

                                int length;

                                while((length = instream.read(buffer)) > 0)
                                {
                                        outstream.write(buffer, 0, length);
                                }

                                instream.close();
                                outstream.close();

                                System.out.println("Copied successfully...");

                    }
                    catch(IOException ioe)
                    {
                                ioe.printStackTrace();
                    }
          }
}
```

OUTPUT :

Cristiano - Notepad

File Edit Format View Help

```
Name: Cristiano Ronaldo
Age: 36
Date of birth: 5 February 1985
Sex: Male
Nationality: Portugal
Professions: Athlete,Footballer
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Player_details - Notepad

File Edit Format View Help

```
Name: Lionel Messi
Age: 34
Date of birth: 24 June 1987
Sex: Male
Nationality: Argentina
Professions: Footballer

Name: Sunil Chhetri
Age: 37
Date of Birth: 3 August 1984
Sex: Male
Nationality: India
Professions: Footballer
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> co6qn3 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.oper
Copied successfully...
```

```
Player_details - Notepad
File  Edit  Format  View  Help
Name: Cristiano Ronaldo
Age: 36
Date of birth: 5 February 1985
Sex: Male
Nationality: Portugal
Professions: Athlete,Footballer
Ln 1, Col 1   100%   Windows (CRLF)   UTF-8
```

PROGRAM 46

AIM

Write a program that reads from a file having integers. Copy even numbers and odd numbers to separate files.

ALGORITHM

Step.1: Start

Step.2: Create a class named ‘co6_q4’.

Step.3: Create an object for the class File to initialize its constructor with the given file.

Step.4: Get user inputs via the console, for the integers to be inserted into the file.

Step.5: Using an object for the FileWriter class, write those integers into the file.

Step.6: Using objects for the FileOutputStream class, create two separate files to store even and odd integers respectively and copy the integers accordingly to separate files just created.

```
while((source.read()) != -1)
{
  if(i%2==0)
  even.write(i);
  else
  odd.write(i);
 }
```

Step 7: Stop

PROGRAM CODE :

co6qn4.java

```
package co6;

import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class co6qn4
{
	public static void main(String args[])throws IOException
	{
		FileInputStream fr = new
FileInputStream("C:\\Users\\ACER\\OneDrive\\Desktop\\Java
Programs\\Integers.txt");
		FileOutputStream fw1 = new
FileOutputStream("C:\\Users\\ACER\\OneDrive\\Desktop\\Java
Programs\\Odd num.txt");
		FileOutputStream fw2 = new
FileOutputStream("C:\\Users\\ACER\\OneDrive\\Desktop\\Java
Programs\\Even num.txt");

		System.out.println(" Successfully copied odd and even
numbers in different files");

		int i;
		while((i=fr.read()) != -1)
		{
			if(i % 2 == 0)
			{
				fw2.write(i);
			}
			else
			{
				fw1.write(i);
			}
		}
		fr.close();
		fw1.close();
		fw2.close();
	}
}
```

OUTPUT :

```
Problems  @ Javadoc  Declaration  Console  Properties
<terminated> co6qn4 [Java Application] C:\Users\ACER\.p2\pool\plugins\org.eclipse.justj.ope
Successfully copied odd and even numbers in different files
```

Integers - Notepad

File Edit Format View Help

0
1
2
3
4
5
6
7
8
9

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Even num - Notepad

File Edit Format View Help

0

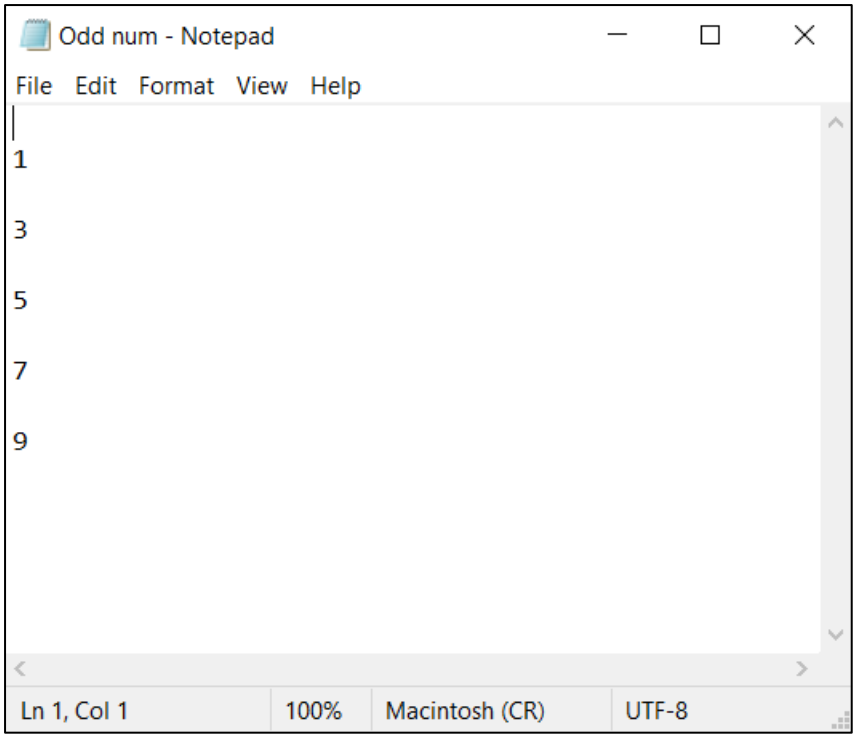
2

4

6

8

Ln 1, Col 1 | 100% | Unix (LF) | UTF-8


Odd num - Notepad
File Edit Format View Help
1
3
5
7
9
Ln 1, Col 1
100%
Macintosh (CR)
UTF-8

PROGRAM 47

AIM :

Client server communication using Socket – TCP/IP

ALGORITHM :

Step.1: Start

Step.2: To create the *Client application*, create an instance of ClientSocket class.

2.1: Initiate connection to the server using hostname and a port number
2.2: Send data to the server using an *OutputStream* object.
2.3: Receive data from the server using an *InputStrem* object.

Step.3: To create the *Server application*, create an instance of ServerSocket class.

3.1: Wait till a connection is established.

Socket s = ss.accept();

3.2: Receive data from the client using an *InputStream* object.
3.3: Send data to the client using an *OutputStream* object.
3.4: Close the connection.

Step.4: Stop

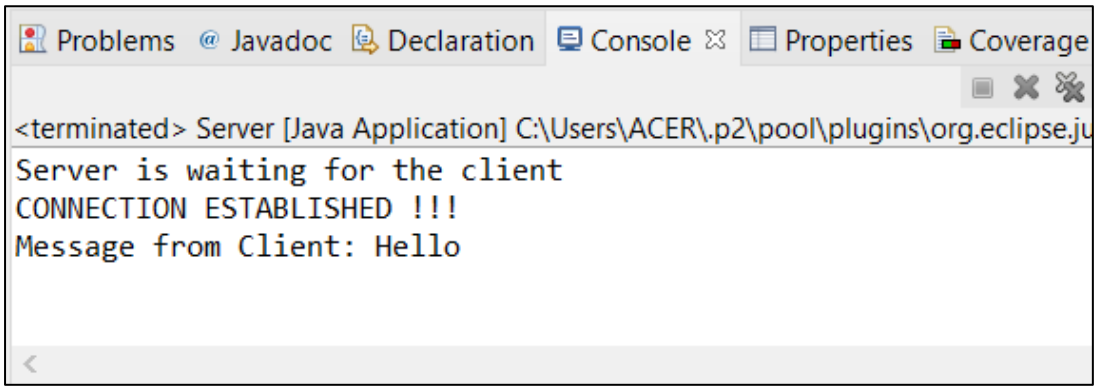
PROGRAM CODE :

Client.java	package CO6;
	import java.net.*;
	import java.io.*;
	
	public class Client {
	public static void main(String args[]) throws Exception{
	try {
	Socket s = new Socket ("localhost", 2665);
	PrintWriter pw = new PrintWriter(s.getOutputStream(), true);
	pw.println("Hello");
	pw.flush();
	
	}
	catch(Exception e) {
	System.out.println("An error occured..." +e);
	}

	} }

server.java	package CO6;
	import java.net.*;
	import java.io.*;
	public class server {
	public static void main(String[] args) throws Exception {
	// TODO Auto-generated method stub
	try {
	ServerSocket ss = new ServerSocket(2665);
	System.out.println("Server is waiting for the client ");
	Socket s = ss.accept();
	System.out.println("CONNECTION ESTABLISHED !!!");
	InputStreamReader isr = new InputStreamReader(s.getInputStream());
	BufferedReader br = new BufferedReader(isr);
	String str = br.readLine();
	System.out.println("Message from Client: "+str);
	//Server is responding through its OutputStream
	PrintWriter pw = new PrintWriter(s.getOutputStream(), true);
	pw.println("Hi Client!! I'm good.");
	pw.close();
	}
	catch(Exception e) {
	System.out.println("An error occured.."+e);
	}
	}
	}

OUTPUT



PROGRAM 48

AIM

Client Server communication using DatagramSocket - UDP

ALGORITHM

Step.1: Start the program.

Step.2: Create the *Client application*:

2.1: Create a *DatagramSocket object* to carry the packet to the destination and to receiveit whenever the server sends any data.

2.2: Create the packet for sending/receiving data via a DatagramSocket

DatagramPacket(byte buf[], int length, InetAddress inetaddress, int port):-

2.3: Invoke a *send()* or *receive()* call on socket object.
2.4: Close the connection.

Step.3: Create the *Server application*:

3.1: Create a *DatagramSocket object* to listen at the port specified.

3.2: Create the packet for sending/receiving data via a DatagramSocket.

3.4: Invoke a *send()* or *receive()* call on socket object.
3.5: Close the connection.

Step.4: Stop the program.

PROGRAM CODE

udpserver.java	package co6;
	import java.io.*;
	import java.net.*;
	
	public class udpserver {
	  public static void main(String[] args) throws IOException {
	  DatagramSocket server=new DatagramSocket(4220);
	  byte[] buf=new byte[256];
	  DatagramPacket packet=new DatagramPacket(buf,buf.length);
	  server.receive(packet);
	  String response =new String(packet.getData());

	System.out.println(" Server : "+response);
	server.close();
	}
	
	}

udpclient.java	package co6;
	
	import java.io.*;
	import java.net.*;
	
	public class udpclient {
	public static void main(String[] args) throws IOException {
	DatagramSocket client= new DatagramSocket();
	InetAddress add=InetAddress.getByName("localhost");
	String str ="Ping from Client!!!";
	byte[] bufBytes = str.getBytes();
	DatagramPacket datagramPacket=new
	DatagramPacket(bufBytes,bufBytes.length,add,4220);
	client.send(datagramPacket);
	client.close();
	}
	
	}

OUTPUT