**Experiment No.: 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.

**CO1:**

Understand object-oriented concepts and design classes and objects to solve problems.

**Procedure:**

import java.util.Scanner;

public class Product{

int pcode;

String pname;

int price;

public void get(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Pcode: ");

pcode=sc.nextInt();

System.out.println("Enter the Product name: ");

pname=sc.next();

System.out.println("Enter the Product Price: ");

price=sc.nextInt();

}

public void put(){

System.out.println("The product details are: ");

System.out.println("The Product code: "+pcode);

System.out.println("The product name: "+pname);

System.out.println("The product price:"+price);

}

public static void main(String[] args){

Product p1=new Product();

Product p2=new Product();

Product p3=new Product();

p1.get();

p2.get();

p3.get();

p1.put();

p2.put();

p3.put();

if(p1.price<p2.price && p1.price<p3.price){

System.out.println("Price of Product 1 is lesser: "+p1.price);

}

else if(p2.price<p1.price && p2.price<p3.price){

System.out.println("Price of Product 2 is lesser: "+p2.price);

}

else{

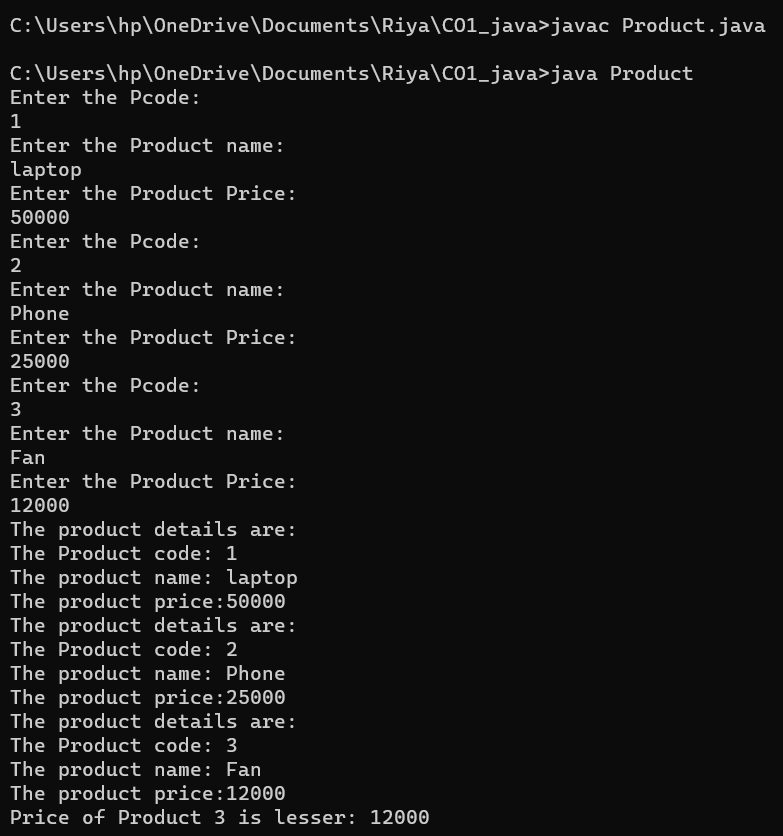
System.out.println("Price of Product 3 is lesser: "+p3.price);

}

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No.: 2**

**Aim:**

Read 2 matrices from the console and perform matrix addition.

**CO1:**

Understand object-oriented concepts and design classes and objects to solve problems.

**Procedure:**

import java.util.\*;

public class Matrixadd{

public static void main(String[] args){

Scanner Snr = new Scanner(System.in);

System.out.println("Enter the dimensions of both matrix: ");

int rw = Snr.nextInt();

int cl = rw;

int Mtx1[][] = new int[rw][cl];

int Mtx2[][] = new int[rw][cl];

int Mtx3[][] = new int[rw][cl];

System.out.println("Enter the elements in first matrix: ");

for(int i=0;i<rw;i++){

for(int j=0;j<cl;j++){

Mtx1[i][j] = Snr.nextInt();

}

}

System.out.println("The First matrix: ");

for(int i=0;i<rw;i++){

for(int j=0;j<cl;j++){

System.out.print(Mtx1[i][j] + "   ");

}

System.out.println(" ");

}

System.out.println("Enter the elements in second matrix: ");

for(int i=0;i<rw;i++){

for(int j=0;j<cl;j++){

Mtx2[i][j] = Snr.nextInt();

}

}

System.out.println("The Second matrix: ");

for(int i=0;i<rw;i++){

for(int j=0;j<cl;j++){

System.out.print(Mtx2[i][j] + "   ");

}

System.out.println(" ");

}

System.out.println("The sum of the matrices: ");

for(int i=0;i<rw;i++){

for(int j=0;j<cl;j++){

Mtx3[i][j] = Mtx1[i][j] + Mtx2[i][j];

}

}

for(int i=0;i<rw;i++){

for(int j=0;j<cl;j++){

System.out.print(Mtx3[i][j] + "   ");

}

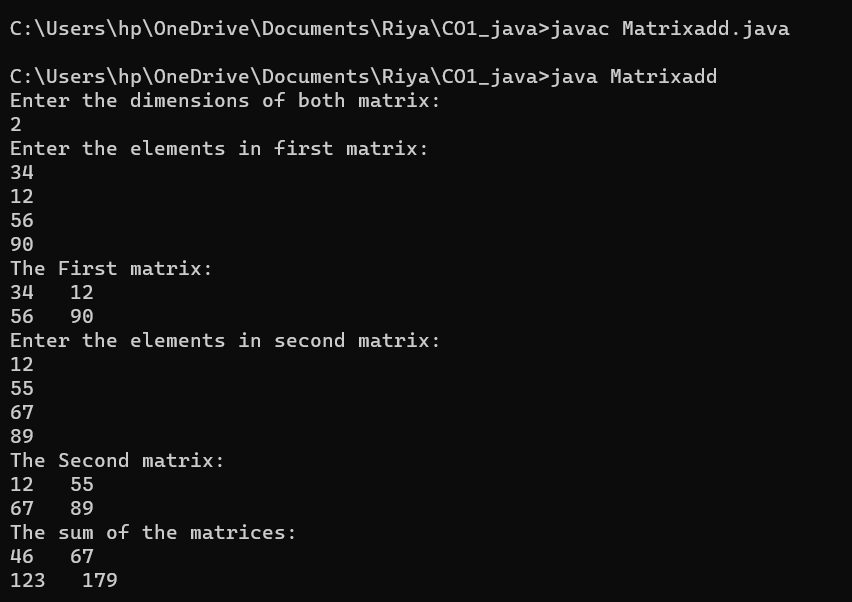
System.out.println(" ");

}

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No.: 3**

**Aim:**

Add complex numbers

**CO1:**

Understand object-oriented concepts and design classes and objects to solve problems.

**Procedure:**

import java.util.Scanner;

public class Complex{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.println("Enter the Real part of first complex number");

int r1=sc.nextInt();

System.out.println("Enter the Imaginary part of first complex number");

int i1=sc.nextInt();

System.out.println("First Complex Number : "+r1+"+"+i1+"i");

System.out.println("Enter the Real part of second complex number");

int r2=sc.nextInt();

System.out.println("Enter the Imaginary part of second complex number");

int i2=sc.nextInt();

System.out.println("First Complex Number : "+r2+"+"+i2+"i");

System.out.println("The sum of Complex numbers are: ");

int r = r1+r2;

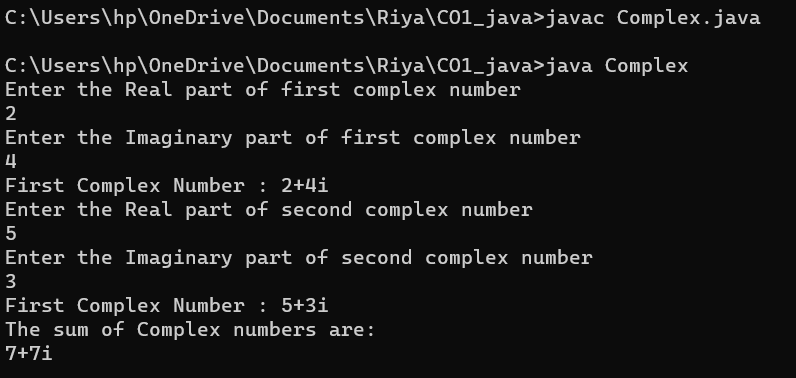
int i = i1+i2;

System.out.println(r+"+"+i+"i");

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No.: 4**

**Aim:**

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

**CO1:**

Understand object-oriented concepts and design classes and objects to solve problems.

**Procedure:**

import java.util.\*;

public class SymmetricMatrix{

public static void main(String[] args){

Scanner Snr = new Scanner(System.in);

System.out.println("Enter the dimension of the matrix :");

int sz = Snr.nextInt();

int Arr[][] = new int[sz][sz];

int Arr1[][] = new int[sz][sz];

System.out.println("Enter the elements in matrix :");

for(int i=0;i<sz;i++){

for(int j=0;j<sz;j++){

Arr[i][j] = Snr.nextInt();

}

}

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

for(int i=0;i<sz;i++){

for(int j=0;j<sz;j++){

System.out.print(Arr[i][j] + " ");

}

System.out.println(" ");

}

System.out.println("The Transpose of the matrix: ");

for(int i=0;i<sz;i++){

for(int j=0;j<sz;j++){

Arr1[i][j]=Arr[j][i];

}

}

for(int i=0;i<sz;i++){

for(int j=0;j<sz;j++){

System.out.print(Arr1[i][j] + " ");

}

System.out.println(" ");

}

int flag=0;

for(int i=0;i<sz;i++){

for(int j=0;j<sz;j++){

if(Arr[i][j] != Arr1[j][i]){

flag=1;

break;

}

}

}

if(flag==1){

System.out.println("The matrix is not symmetric");

}

else{

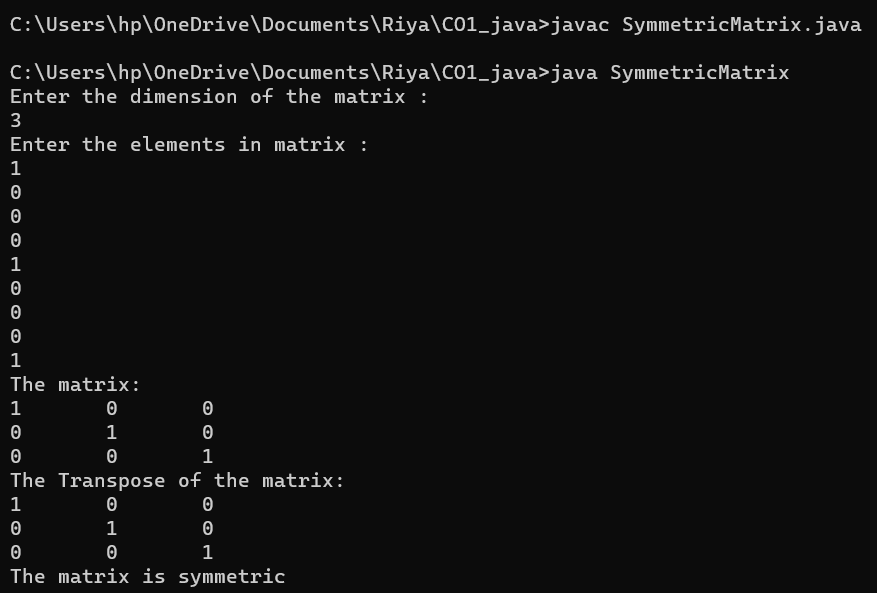
System.out.println("The matrix is symmetric");

}

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO1 was obtained.

**Experiment No.: 5**

**Aim:**

Program to Sort strings

**CO2:**

Familiarization and understanding of arrays and strings

**Procedure:**

import java.util.Scanner;

public class Compare{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the Array ");

int l =sc.nextInt();

String str[] = new String[l];

int i;

System.out.println("Enter the elements for array: ");

for(i=0;i<str.length;i++){

str[i]=sc.next();

}

System.out.println("The array : ");

for(i=0;i<str.length;i++){

System.out.println(str[i]);

}

String temp;

int j;

for(i=0;i<str.length;i++){

for(j=i+1;j<str.length;j++){

if(str[i].compareTo(str[j])>0){

temp=str[i];

str[i]=str[j];

str[j]=temp;

}

}

}

System.out.println("The Sorted Array : ");

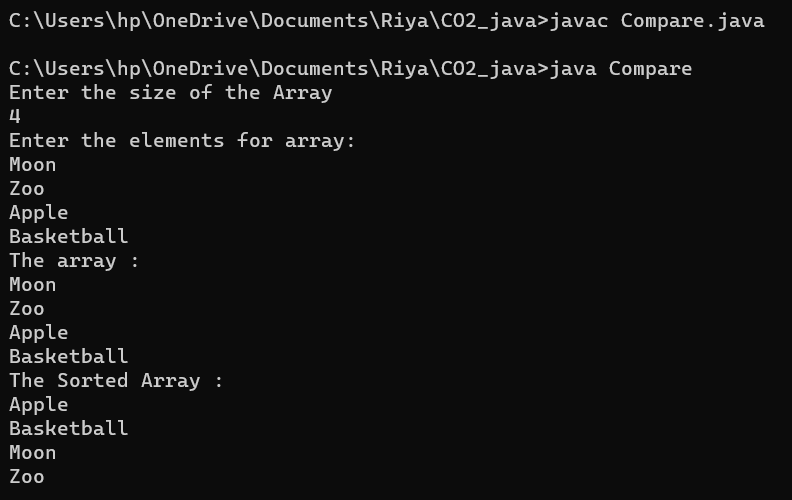
for(i=0;i<str.length;i++){

System.out.println(str[i]);

}

}}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

**Experiment No.: 6**

**Aim:**

Search an element in an array.

**CO2:**

Familiarization and understanding of arrays and strings

**Procedure:**

import java.util.Scanner;

public class Array{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.println("Enter the size of Array: ");

int l=sc.nextInt();

int num[]=new int[l];

int i;

System.out.println("Enter the elements for array");

for(i=0;i<num.length;i++){

num[i]=sc.nextInt();

}

System.out.println("The array is");

for(i=0;i<num.length;i++){

System.out.println(num[i]);

}

//search

System.out.println("Enter the number to be searched: ");

int val=sc.nextInt();

for(i=0;i<num.length;i++){

if(num[i]==val){

System.out.println("The value found");

}

else{

System.out.println("The value not found");

break;

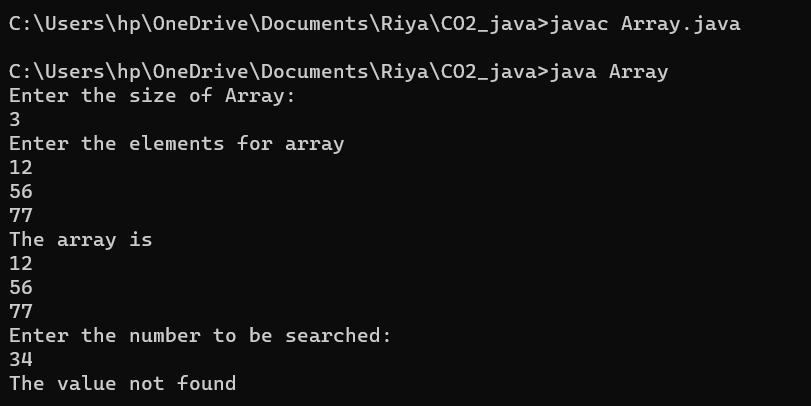
}

}

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

**Experiment No.: 7**

**Aim:**

Perform string manipulations

**CO2:**

Familiarization and understanding of arrays and strings

**Procedure:**

import java.util.Scanner;

public class Manipulate{

public static void main(String[] args){

String str1,str2;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the first string: ");

str1=sc.nextLine();

System.out.println("Enter the second string: ");

str2 =sc.nextLine();

System.out.println("Concate: "+str1+" "+str2);

System.out.println("Lower Case: "+str1.toLowerCase());

System.out.println("Upper Case: "+str1.toUpperCase());

System.out.println("The length of the String 2 is : "+str2.length());

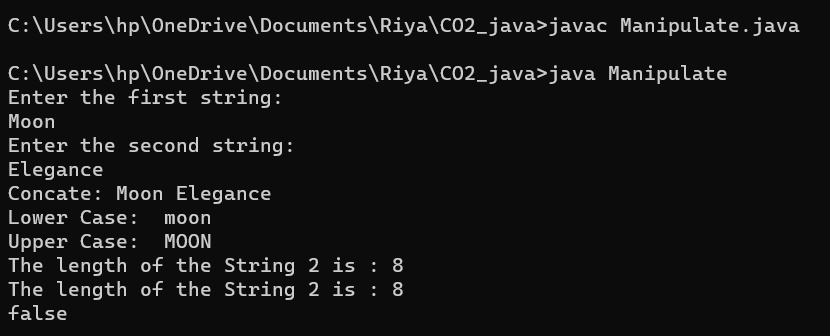
System.out.println("The length of the String 2 is : "+str2.length());

System.out.println(str1.equals(str2);

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

**Experiment No.: 8**

**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.

**CO2:**

Familiarization and understanding of arrays and strings

**Procedure:**

import java.util.Scanner;

public class Employee{

int eNo;

String eName;

int eSalary;

public void get(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Employee Number: ");

eNo=sc.nextInt();

System.out.println("Enter the Employee Name: ");

eName=sc.next();

System.out.println("Enter the Employee Salary: ");

eSalary=sc.nextInt();

}

public void put(){

//System.out.println(" ");

System.out.println("The Employee details are: ");

System.out.println("The Employee Number: "+eNo);

System.out.println("The Employee Name : "+eName);

System.out.println("The Employee Salary :"+eSalary);

}

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int n;

int i;

int val;

System.out.println("Enter the number of Employees: ");

n=sc.nextInt();

Employee e[]=new Employee[n];

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

e[i]=new Employee();

e[i].get();

}

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

e[i].put();

}

System.out.println("Enter the Employee number to be searched: ");

val=sc.nextInt();

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

if(e[i].eNo==val){

e[i].put();

break;

}

else

{

System.out.println("Data Not Found!!");

break;

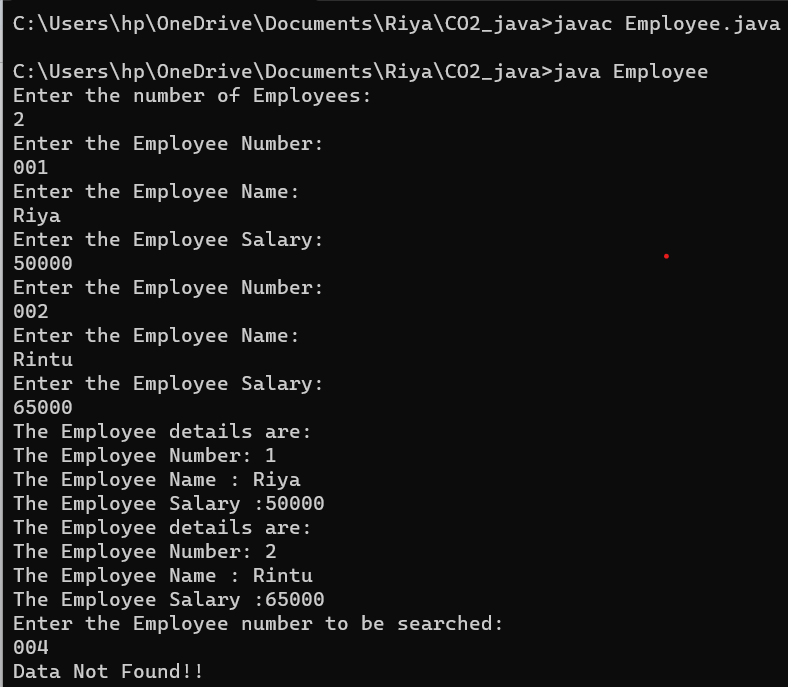
}

}

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO2 was obtained.

**Experiment No.: 9**

**Aim:**

Area of different shapes using overloaded functions

**CO3:**

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure:**

import java.util.\*;

class Shapes{

int r,h,a,p1,p2,t2,t3;

double b,l,w,t1;

public void area(double r){

double c=3.14\*(r\*r);

System.out.println("Area of circle: "+c);

}

public void area(double b,int h){

double t=(b\*h)/2;

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

}

public void area(int a){

int s=a\*a;

System.out.println("Area of square: "+s);

}

public void area(double l,double w){

double r=l\*w;

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

}

public void area(int p1,int p2){

int p=p1\*p2;

System.out.println("Area of parallelogram: "+p);

}

public void area(double t1,int t2,int t3){

double tr=0.5\*(t2+t3)\*t1;

System.out.println("Area of trapezium: "+tr);

}

public void area(float e1,float e2){

double c=3.14\*(e1\*e2);

System.out.println("Area of ellipse: "+c);

}}

public class Shape{

public static void main(String[] args){

int ch;

Scanner s=new Scanner(System.in);

Shapes sh =new Shapes();

System.out.println("Area of different shapes"+"\n"+" 1.Circle"+"\n"+" 2.Triangle"+"\n"+" 3.Square "+"\n"+" 4.Rectangle "+"\n"+" 5.Parallelogram "+"\n"+" 6.Trapezium"+"\n"+" 7.Ellipse");

do{

System.out.println("Enter choice: ");

ch=s.nextInt();

switch(ch){

case 1:

{

System.out.println("Enter the radius: ");

double r=s.nextDouble();

sh.area(r);

}

break;

case 2:

{

System.out.println("Enter the breadth: ");

double b=s.nextDouble();

System.out.println("Enter the height: ");

int h=s.nextInt();

sh.area(b,h);

}

break;

case 3:

{

System.out.println("Enter the length: ");

int a=s.nextInt();

sh.area(a);

}

break;

case 4:

{

System.out.println("Enter the length: ");

double l=s.nextDouble();

System.out.println("Enter the breadth: ");

double w=s.nextDouble();

sh.area(l,w);

}

break;

case 5:

{

System.out.println("Enter the base: ");

int p1=s.nextInt();

System.out.println("Enter the vertical height: ");

int p2=s.nextInt();

sh.area(p1,p2);

}

break;

case 6:

{

System.out.println("Enter the height of trapezium: ");

double t1=s.nextDouble();

System.out.println("Enter the length of parallel side: ");

int t2=s.nextInt();

System.out.println("Enter the length of parallel side: ");

int t3=s.nextInt();

sh.area(t1,t2,t3);

}

break;

case 7:

{

System.out.println("Enter the minor axis: ");

float e1=s.nextFloat();

System.out.println("Enter the major axis: ");

float e2=s.nextFloat();

sh.area(e1,e2);

}

break;

/\*default:

System.out.println("invalid choice");

break;\*/

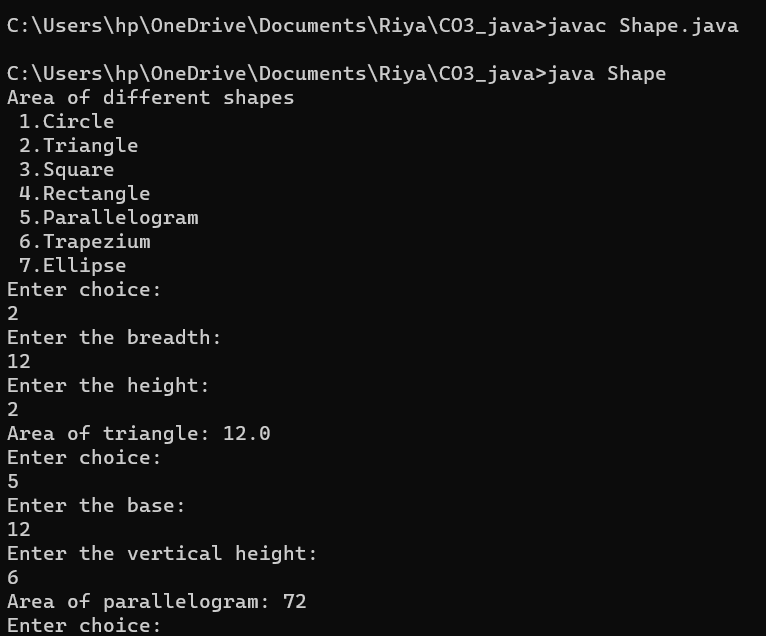
}}

while(ch!=7);

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No.: 10**

**Aim:**

Create a class ‘Employee’ 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.

**CO3:**

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure:**

import java.util.Scanner;

class Employee{

int empid;

String Ename;

int Esalary;

String Eaddress;

Employee(int id, String name, int salary , String Add){

empid=id;

Ename=name;

Esalary=salary;

Eaddress=Add;

}}

class Teacher extends Employee{

String department;

String sub\_taught;

Teacher(int id, String name, int salary , String Add,String dept, String sub){

super(id,name,salary,Add);

department=dept;

sub\_taught=sub;

}

void show(){

System.out.println("The Employee id:"+empid);

System.out.println("The Employee name:"+Ename);

System.out.println("The Employee salary:"+Esalary);

System.out.println("The Employee address:"+Eaddress);

System.out.println("The Employee department:"+department);

System.out.println("The Subject taught:"+sub\_taught);

}}

public class EmpTeacher{

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

int i;

System.out.println("Enter the no. of Employees:");

int l =sc.nextInt();

Teacher t[]=new Teacher[l];

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

System.out.println("Enter the employee id:");

int empid=sc.nextInt();

System.out.println("Enter the employee name:");

String empname= sc.next();

System.out.println("Enter the employee salary:");

int empsal=sc.nextInt();

System.out.println("Enter the employee address:");

String empaddress=sc.next();

System.out.println("Enter the employee department:");

String empdept=sc.next();

System.out.println("Enter the Subject taught:");

String empsub=sc.next();

t[i]= new Teacher(empid,empname,empsal,empaddress,empdept,empsub);

}

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

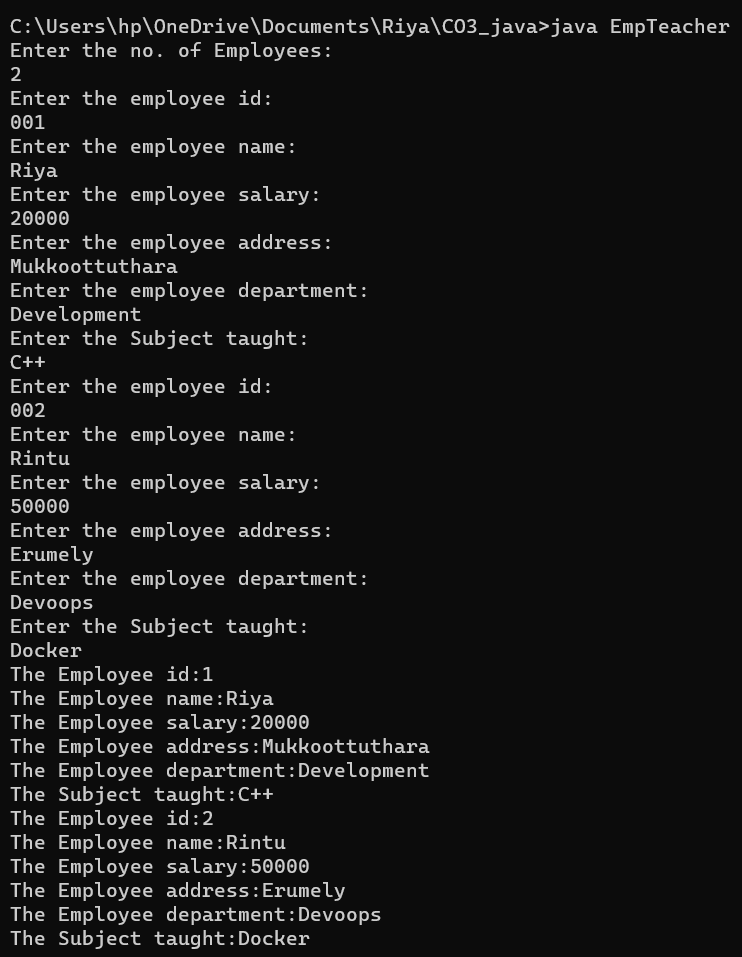
t[i].show();

}

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No.: 11**

**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.

**CO3:**

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure:**

import java.util.\*;

class Person{

String Pname;

String Pgender;

String Paddress;

int Page;

Person(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Name:");

Pname=sc.next();

System.out.println("Enter the Gender:");

Pgender=sc.next();

System.out.println("Enter the Address:");

Paddress=sc.next();

System.out.println("Enter the Age:");

Page=sc.nextInt();

}

}

class Employee extends Person{

int empid;

String company\_name;

String qualification;

int salary;

Employee(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Employee id:");

empid=sc.nextInt();

System.out.println("Enter the Company name:");

company\_name=sc.next();

System.out.println("Enter the Qualification:");

qualification=sc.next();

System.out.println("Enter the Salary");

salary=sc.nextInt();

}}

class Teacher extends Employee{

String subject;

String Department;

int Teacherid;

Teacher(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Subject:");

subject=sc.next();

System.out.println("Enter the Department:");

Department=sc.next();

System.out.println("Enter the Teacher id:");

Teacherid=sc.nextInt();

System.out.println("---------------------------------------");

}

public void show(){

System.out.println("##########");

System.out.println("The Name: "+Pname);

System.out.println("The Gender:"+Pgender);

System.out.println("The Address:"+Paddress);

System.out.println("The Age:"+Page);

System.out.println("The Employee id: "+empid);

System.out.println("The Company name:"+company\_name);

System.out.println("The Qualification:"+qualification);

System.out.println("The Salary:"+salary);

System.out.println("The Teacher id :"+Teacherid);

System.out.println("The Subject:"+subject);

System.out.println("The Department:"+Department);

System.out.println(" ");

}

}

public class Second{

public static void main(String[] args){

Scanner sc = new Scanner(System.in);

int i;

System.out.println("Enter the no. of Employees:");

int l =sc.nextInt();

Teacher t[]=new Teacher[l];

System.out.println("Enter the Details of :"+l+" Emplyess are: ");

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

t[i]= new Teacher();

}

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

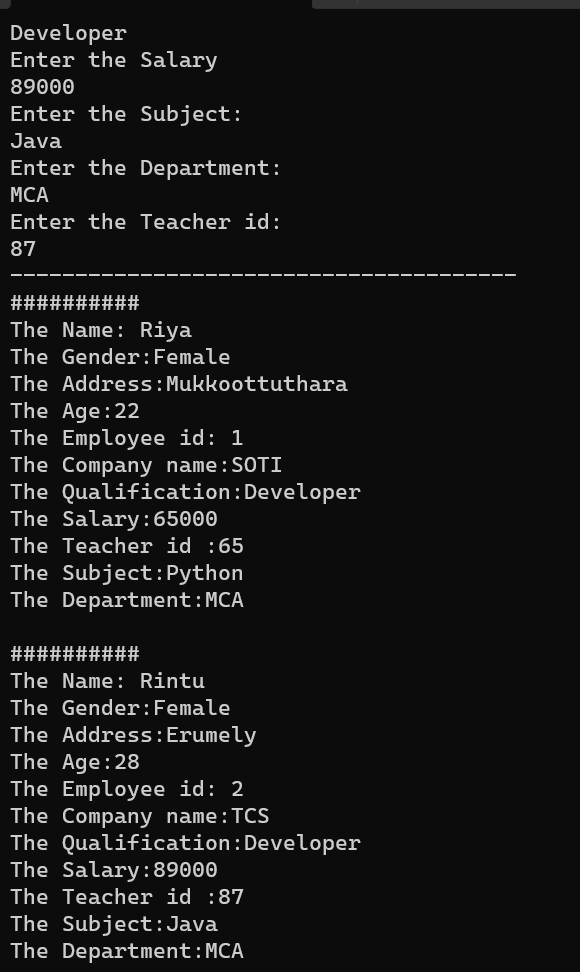
t[i].show();

}

}

}

**Output Screenshot:**

 ****

**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No.: 12**

**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.

**CO3:**

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure:**

import java.util.Scanner;

class Publisher{

String Pubname;

Publisher(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the publisher: ");

Pubname=sc.next();

} }

class books extends Publisher{

String booktype ;

books(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the type of the book Literature or Fiction: ");

booktype= sc.next();

} }

class Literature extends books{

String title , author ;

int price;

Literature(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the title of the literature book: ");

title= sc.next();

System.out.println("Enter the author of the literature book: ");

author= sc.next();

System.out.println("Enter the price of the literature book: ");

price= sc.nextInt();

}

void display(){

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("The Details of the Literature Book!!");

System.out.println("Publisher Name: "+Pubname);

System.out.println("Type of the Book: "+booktype );

System.out.println("Title: "+title);

System.out.println("Author: "+author);

System.out.println("Price: "+price);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" ");

} }

class Fiction extends books{

String title1 , author1 ;

int price1;

Fiction(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the title of the Fiction book: ");

title1= sc.next();

System.out.println("Enter the author of the Fiction book: ");

author1= sc.next();

System.out.println("Enter the price of the Fiction book: ");

price1= sc.nextInt();

}

void display1(){

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println("The Details of the Fiction Book!!");

System.out.println("Publisher Name: "+Pubname);

System.out.println("Type of the Book: "+booktype );

System.out.println("Title: "+title1);

System.out.println("Author: "+author1);

System.out.println("Price: "+price1);

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.println(" ");

} }

public class Book{

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

int i;

System.out.print("Enter number of entries for Literature:");

int ln=sc.nextInt();

Literature l[]=new Literature[ln];

for(i=0;i<ln;i++)

{

l[i]=new Literature();

}

for(i=0;i<ln;i++)

{

l[i].display();

}

System.out.print(" ");

System.out.print("Enter number of entries for Fiction:");

int fn=sc.nextInt();

Fiction f[]=new Fiction[fn];

for(i=0;i<fn;i++)

{

f[i]=new Fiction();

}

for(i=0;i<fn;i++)

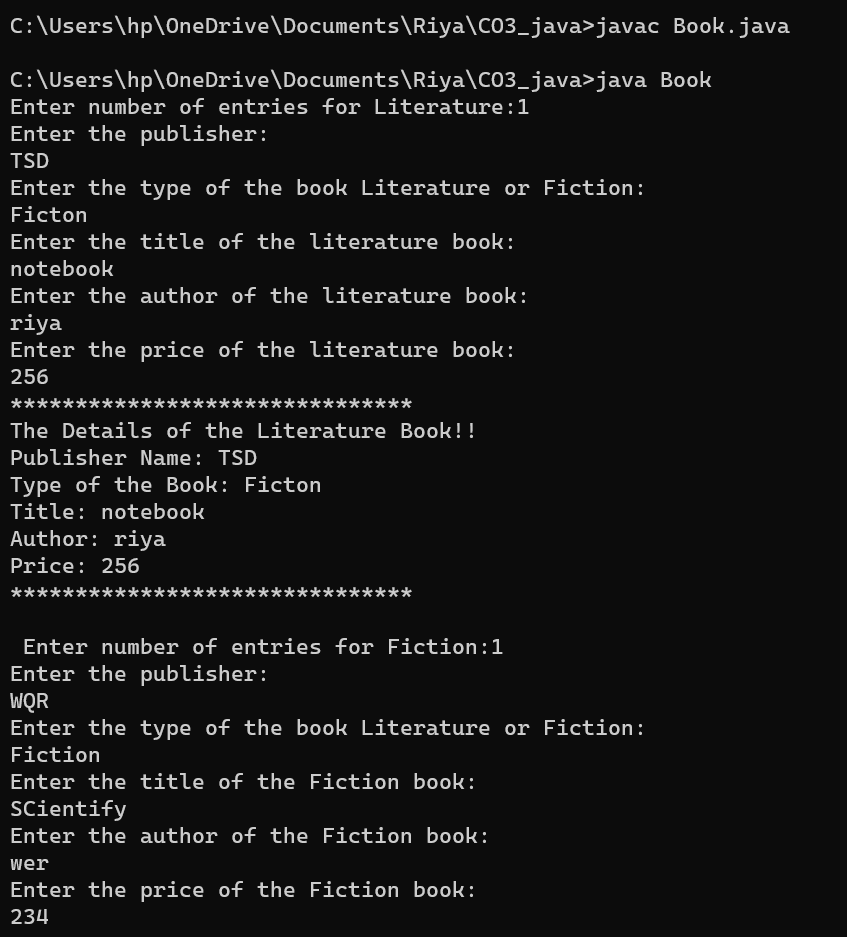
{

f[i].display1();

}

} }

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No.: 13**

**Aim:**

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

**CO3:**

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure:**

import java.util.Scanner;

interface Student{

public void student\_get();

public void student\_display();

}

interface Sports{

public void sport\_get();

public void sport\_display();

}

class Result implements Student,Sports{

String name , sports ;

double rollno , mark1, mark2, mark3,total , pos;

double percentage;

public void student\_get(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the name of the student: ");

name = sc.next();

System.out.println("Enter the Roll No of the student: ");

rollno = sc.nextDouble();

System.out.println("Enter the Mark 1 of the student: ");

mark1 = sc.nextDouble();

System.out.println("Enter the Mark 2 of the student: ");

mark2 = sc.nextDouble();

System.out.println("Enter the Mark 3 of the student: ");

mark3 = sc.nextDouble();

total = mark1+mark2+mark3;

percentage = total/300\*100;

}

public void student\_display(){

System.out.println("The Student name: "+name);

System.out.println("Roll No.: "+rollno);

System.out.println("Total marks.: "+total);

System.out.println("Percentage: "+percentage);

}

public void sport\_get(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the sports activity participated: ");

sports = sc.next();

System.out.println("Enter the position scored:");

pos = sc.nextInt();

}

public void sport\_display(){

System.out.println("Activity Participated: "+sports);

System.out.println("Position scored: "+pos);

}

}

public class Result1{

public static void main(String[] args){

Result obj = new Result();

obj.student\_get();

obj.sport\_get();

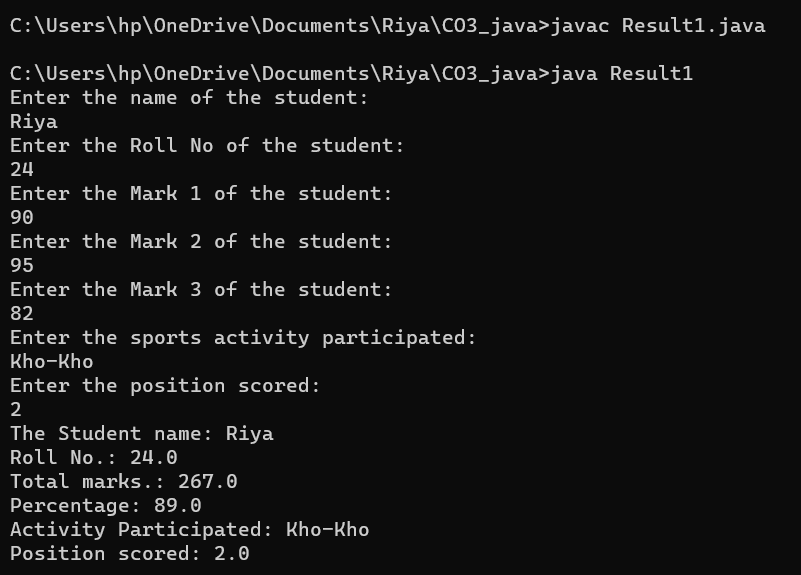
obj.student\_display();

obj.sport\_display();

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No.: 14**

**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.

**CO3:**

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure:**

import java.util.Scanner;

interface prototype{

public void get\_data();

public void display();

public void area();

public void perimeter();

}

class Circle implements prototype{

int r ;

double area,perimeter;

public void get\_data(){

Scanner sc = new Scanner(System.in);

System.out.println("Enter the radius of the circle: ");

r=sc.nextInt();

}

public void area(){

area=3.14\*r\*r;

}

public void perimeter(){

perimeter=2\*3.14\*r;

}

public void display(){

System.out.println(" ");

System.out.println("------CIRCLE-------");

System.out.println("Area of Circle: "+area+" sq.units");

System.out.println("Perimeter of Circle: "+perimeter+" units");

}

}

class Rectangle implements prototype{

int l,b,area1,perimeter1;

public void get\_data(){

System.out.println(" ");

Scanner sc = new Scanner(System.in);

System.out.println("Enter the length of the rectangle: ");

l=sc.nextInt();

System.out.println("Enter the breadth of the rectangle: ");

b=sc.nextInt();

}

public void area(){

area1=l\*b;

}

public void perimeter(){

perimeter1=2\*(l+b);

}

public void display(){

System.out.println(" ");

System.out.println("------Rectangle-------");

System.out.println("Area of Rectangle: "+area1+" sq.units");

System.out.println("Perimeter of Rectangle: "+perimeter1+" units");

}

}

public class Shapes{

public static void main(String[] args){

int ch,u=0;

Scanner sc = new Scanner(System.in);

Circle obj =new Circle();

Rectangle obj1=new Rectangle();

do{

System.out.println("\n1.Circle\n2.Rectangle\n3.EXIT ");

System.out.println("-------------------------------------- ");

System.out.println("Enter your Choice:");

ch=sc.nextInt();

switch(ch){

case 1: obj.get\_data();

obj.area();

obj.perimeter();

obj.display();

break;

case 2:obj1.get\_data();

obj1.area();

obj1.perimeter();

obj1.display();

break;

case 3:System.out.println("EXITED");

System.exit(0);

}

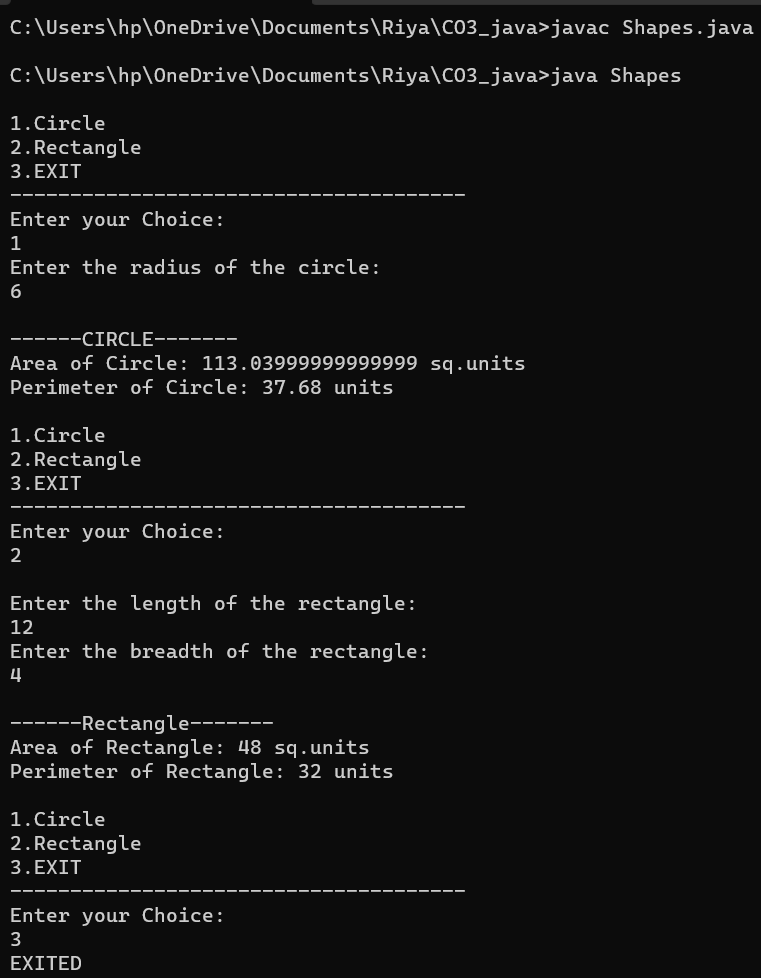
}

while(ch!=3);

}

}

**Output Screenshot:**



**Result:**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No.: 15**

**Aim:**

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

Order No.

Date :

ProductId Name Quantity unitprice Total

101 A 2 25 50

102 B 1 100 100

Net.Amount 150

**CO3:**

Understand and implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure:**

import java.util.\*;

interface calculate{

public void calc();

}

class bill implements calculate{

String date,name;

int qu,id;

float uprice,total;

Scanner sc= new Scanner(System.in);

public void get(){

System.out.println("Enter product id");

id=sc.nextInt();

System.out.println("Enter product name");

name=sc.next();

System.out.println("Enter the quantity of the product");

qu=sc.nextInt();

System.out.println("Enter product unit price");

uprice=sc.nextFloat();

}

public void calc(){

total=qu\*uprice;

}

public void disp(){

System.out.println(id+" "+name+" "+qu+" "+uprice+" "+total);

}

}

public class calculate1{

public static void main(String[] args){

int n,i,o;

float net=0;

String date;

Scanner sc= new Scanner(System.in);

System.out.println("Enter order number");

o=sc.nextInt();

System.out.println("Enter order date");

date=sc.next();

System.out.println("Enter no.of products");

n=sc.nextInt();

bill b[]=new bill[n];

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

{

b[i]=new bill();

b[i].get();

b[i].calc();

}

System.out.println("............BILL............");

System.out.println("Order no.:"+o);

System.out.println("Date:"+date);

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

System.out.println("ID NAME QUNTITY PRICE");

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

{

b[i].disp();

net=net+b[i].total;

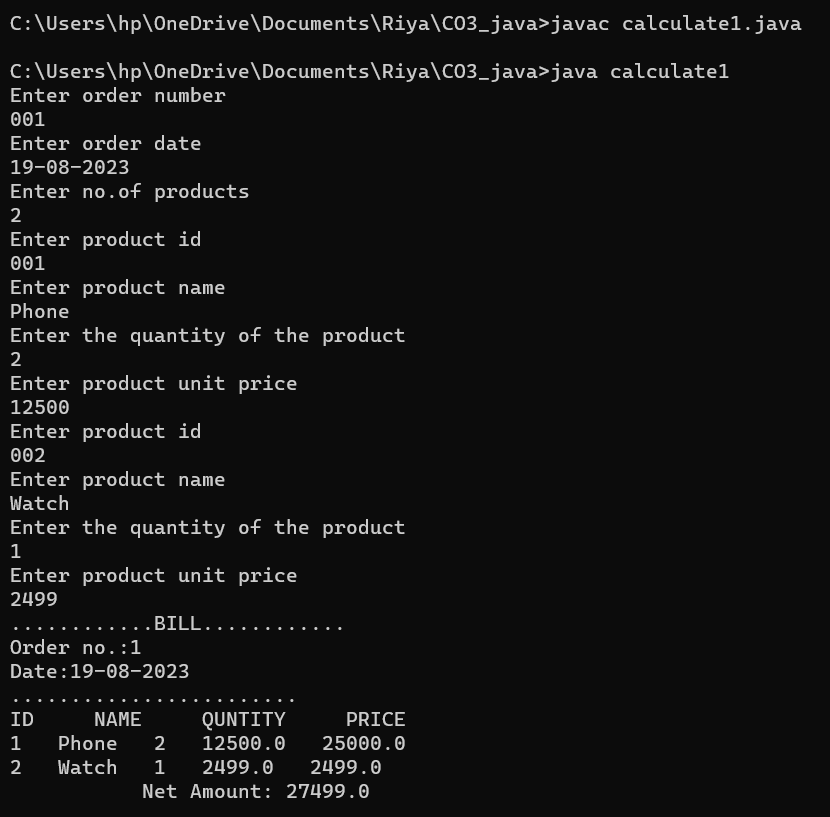
}

System.out.println(" Net Amount: "+net);

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO3 was obtained.

**Experiment No.: 16**

**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.

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

package Graphiccs;

interface Area1 {

public void Rectangle();

public void Triangle();

public void Square();

public void Circle();

public void getRect();

public void getTri();

public void getSqr();

public void getCrl();

}

 //shapes.java

package Graphiccs;

import java.util.\*;

public class Shapes1 implements Area1 {

int lr,lb,ra,th,tb,ta,saa,sa,cr,cc;

public void getRect() {

Scanner ab= new Scanner(System.in);

System.out.println("Enter the length of the rectangle");

lr=ab.nextInt();

System.out.println("Enter the breadth of the rectangle");

lb=ab.nextInt();

}

public void Rectangle(){

ra=lr\*lb;

System.out.println("Area of Rectangle is "+ra);

}

public void getTri(){

Scanner cb= new Scanner(System.in);

System.out.println("Enter the height of the Triangle");

th=cb.nextInt();

System.out.println("Enter the base of the Triangle");

tb=cb.nextInt();

}

public void Triangle(){

ta=0.5\*th\*tb;

System.out.println("Area of Triangle angle is "+ta);

}

public void getSqr(){

Scanner sq= new Scanner(System.in);

System.out.println("Enter the Side of the Square");

sa=sq.nextInt();

}

public void Square(){

saa=sa\*sa;

System.out.println("Area of Square is "+saa);

}

public void getCrl(){

Scanner sc= new Scanner(System.in);

System.out.println("Enter the radius of the Circle");

cc=sc.nextInt();

}

public void Circle(){

cr=3.14\*cc\*cc;

System.out.println("Area of Square is "+cr);

}

public static void main(String[] args){

Shapes1 o= new Shapes1();

o.getRect();

o.Rectangle();

o.getTri();

o.Triangle();

o.getSqr();

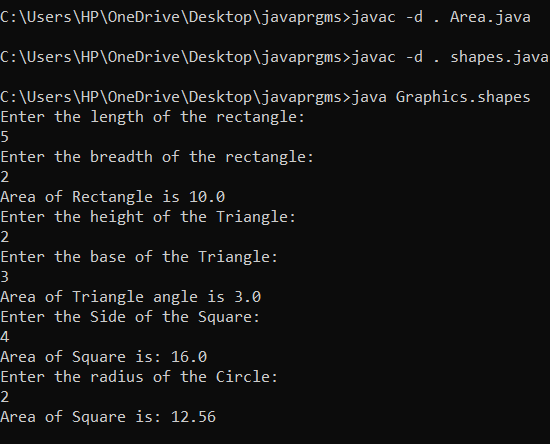
o.Square();

o.getCrl();

o.Circle();

} }

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 17**

**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.

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

package Arithematic;

interface operations{

public void input();

public void add();

public void substract();

public void multiply();

public void division();

}

//basic.java

package Arithematic;

import java.util.\*;

public class basic implements operations{

double a,b,ad,dif,mult,div;

public void input(){

Scanner ab=new Scanner(System.in);

System.out.println("Enter two numbers");

a=ab.nextInt();

b=ab.nextInt();

}

public void add(){

ad=a+b;

System.out.println("Sum is "+ad); }

public void substract(){

dif=a-b;

System.out.println("Difference is "+dif); }

public void multiply(){

mult=a\*b;

System.out.println("Product is "+mult); }

public void division(){

div=a/b;

System.out.println("Quotient is "+div); }

public static void main(String[] args){

basic o=new basic();

o.input();

o.add();

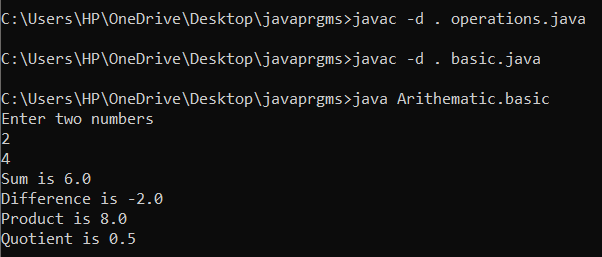
o.substract();

o.multiply();

o.division();

}}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 18**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.Scanner;

class UsernameException extends Exception{

public UsernameException(String msg){

 super(msg);

}

}

class PasswordException extends Exception{

public PasswordException(String msg){

 super(msg);

}

}

public class CheckLoginCredential{

public static void main(String[] args){

 Scanner s = new Scanner(System.in);

 String username, password;

 System.out.print("Enter username :: ");

 username = s.nextLine();

 System.out.print("Enter password :: ");

 password = s.nextLine();

 int length = username.length();

 try {

 if(length < 6)

 throw new UsernameException("Username must be greater than 6 characters???");

 else if(!password.equals("hello"))

 throw new PasswordException("Incorrect password\nType correct password???");

 else

 System.out.println("Login Successful !!!");

 }

 catch (UsernameException u){

u.printStackTrace();

 }

 catch (PasswordException p) {

 p.printStackTrace();

 }

 finally {

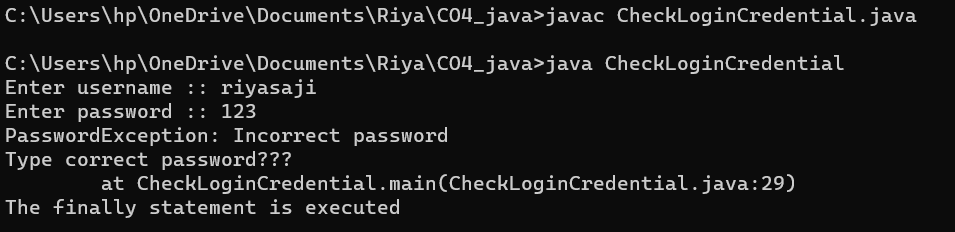
 System.out.println("The finally statement is executed");

 }

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 19**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.\*;

class NegativeNumberException extends Exception {

public NegativeNumberException(String message) {

super(message);

}

}

class AvgException {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of integers: ");

int n = scanner.nextInt();

int sum = 0;

int count = 0;

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

try {

System.out.print("Enter a positive integer: ");

int num = scanner.nextInt();

if (num < 0) {

throw new NegativeNumberException("Negative numbers are not allowed");

}

sum += num;

count++;

}

catch (NegativeNumberException e) {

System.out.println(e.getMessage());

}

}

if (count == 0) {

System.out.println("No positive numbers were entered");

}

else {

double average = (double) sum / count;

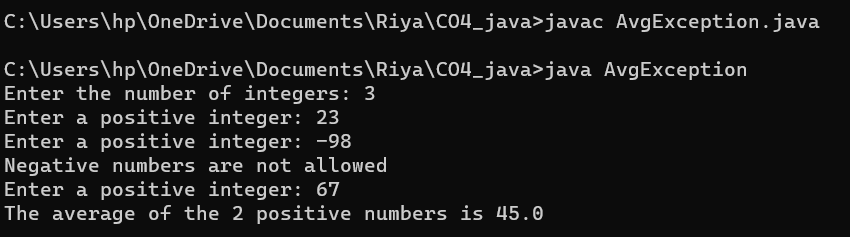
System.out.println("The average of the " + count + " positive numbers is " + average);

}

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 20**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.Scanner;

class Multi extends Thread{

public void run(){

int num=5;

System.out.printf("\_\_\_MULTIPLICATION TABLE OF 5\_\_");

for(int i=1;i<=10;i++){

System.out.printf("%d \* %d = %d\n",num,i,num\*i);

}

}

}

class Prime extends Thread{

public void run(){

int i,j,flag;

Scanner sc=new Scanner(System.in);

System.out.println("\_\_\_TO GENERATE First N PRIME NUMBERS\_\_");

System.out.println("Enter the Limit");

int N=sc.nextInt();

System.out.println("Prime numbers between 1 and"+N+" are: ");

for( i=1;i<=N;i++){

if(i==0 || i==1)

continue;

flag=1;

for(j=2;j<=i/2;j++){

if(i%j==0){

flag=0;

break;

}}

if(flag==1)

System.out.println(i+" ");

}

} }

public class ThreatC{

public static void main(String[] args) throws InterruptedException{

Multi a = new Multi();

a.start();

a.sleep(200);

Prime b=new Prime();

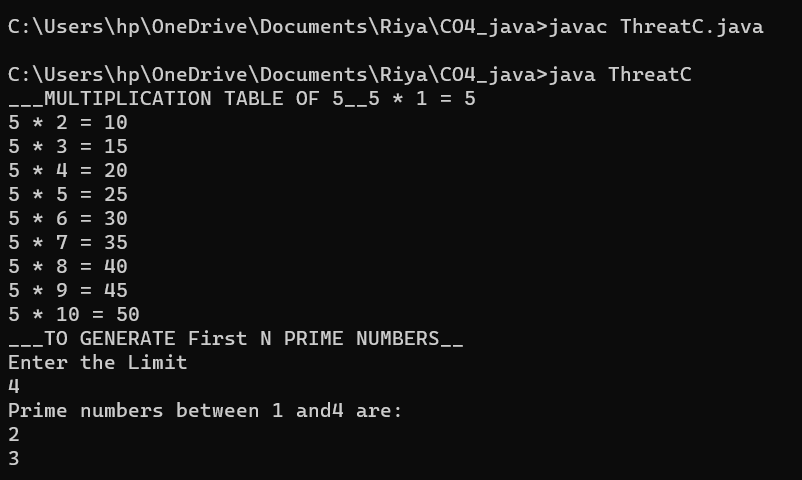
b.start();

b.sleep(200);

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 21**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.Scanner;

class Fibonacci implements Runnable{

public void run(){

int first=0,second=1,next=0;

Scanner sc= new Scanner(System.in);

System.out.println("\_\_\_TO GENERATE FIBONACCI SERIES\_\_");

System.out.println("Enter the number of terms required ");

int n =sc.nextInt();

System.out.println("Series generated");

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

System.out.println(first+" ");

next=first+second;

first=second;

second=next;

}

}

}

class EvenNo implements Runnable{

public void run(){

int lower , upper;

Scanner sc=new Scanner(System.in);

System.out.println("\_\_\_TO GENERATE EVEN NUMBERS IN A GIVEN RANGE\_\_");

System.out.println("Enter the Lower Limit");

lower=sc.nextInt();

System.out.println("Enter the Upper Limit");

upper=sc.nextInt();

System.out.println(" ");

System.out.println("The numbers are: ");

for(int i=lower;i<=upper;i++){

if(i%2==0){

System.out.println(i+" ");

i++;

}

else{

break;

}

}

}

}

public class ThreadR{

public static void main(String[] args) throws InterruptedException{

Fibonacci obj = new Fibonacci();

Thread a =new Thread(obj);

a.start();

a.sleep(2000);

EvenNo obj1=new EvenNo();

Thread b=new Thread(obj1);

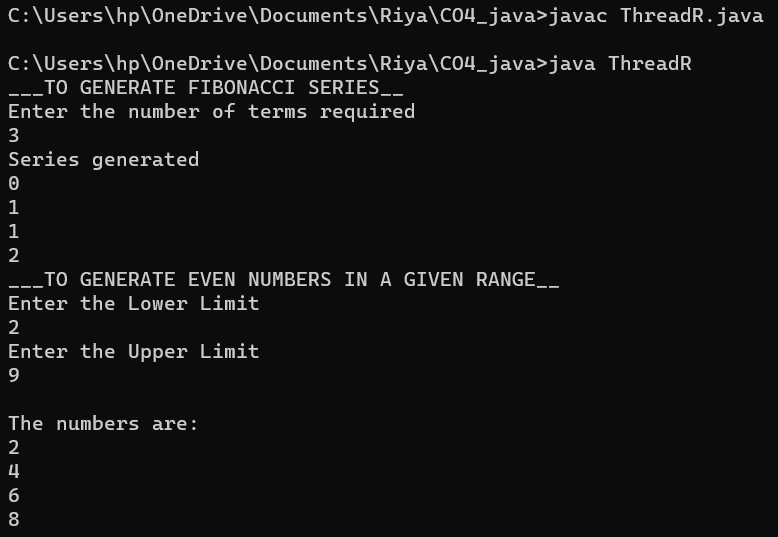
b.start();

b.sleep(1000);

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 22**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.\*;

class stack1

{

int top=-1,n=10,i;

int a[]=new int[n];

void push() {

if(top==n-1)

{

System.out.println("STACK OVERFLOW");

}

else

{

Scanner sc = new Scanner(System.in);

System.out.println("Enter value:");

i=sc.nextInt();

top=top+1;

a[top]=i;

}

}

void pop() {

if(top==-1)

{

System.out.println("STACK UNDERFLOW");

}

else {

top=top-1;

System.out.println("ITEM DELETED");

}

}

void display() {

System.out.println("STACK ELEMENTS ARE: ");

for(int i=top;i>=0;i--)

{

System.out.println(a[i]);

}

}

public static void main(String[] args) {

int n=10,ch;

Scanner sc = new Scanner(System.in);

stack1 s=new stack1();

do{

System.out.println("\*\*\*Stack operation\*\*\*");

System.out.println("1.push 2.pop 3.display 4.exit");

System.out.println("Enter choice:");

ch=sc.nextInt();

switch (ch)

{

case 1:

{

s.push();

break;

}

case 2:

{

s.pop();

break;

}

case 3:

{

s.display();

break;

}

default:

{

System.out.println("enter valid choice");

}

}

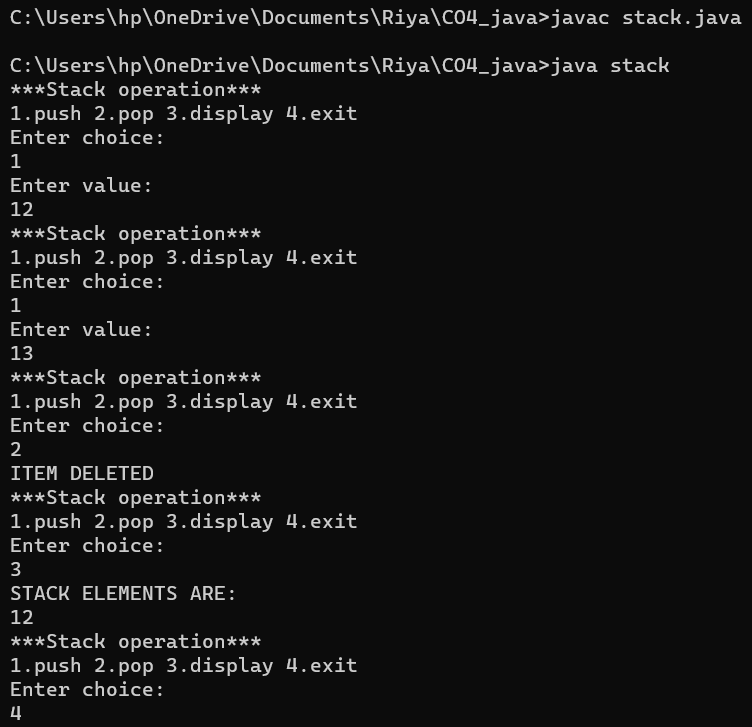
}

while(ch !=0);

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 23**

**Aim:**

Using generic method perform Bubble sort.

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.\*;

class bbsrt{

public static void main(String [] args) {

Scanner sc= new Scanner(System.in);

int i,j;

bbsrt obj=new bbsrt();

System.out.println("Enter the size of the array");

int n=sc.nextInt();

int arr[]=new int[n];

System.out.println("Enter the element");

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

{

arr[i]=sc.nextInt();

}

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

{

for(j=i+1;j<n;j++)

{

if(arr[i]>arr[j])

{

int temp=arr[i];

arr[i]=arr[j];

arr[j]=temp;

}

}

}

System.out.println("inserted elements are");

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

{

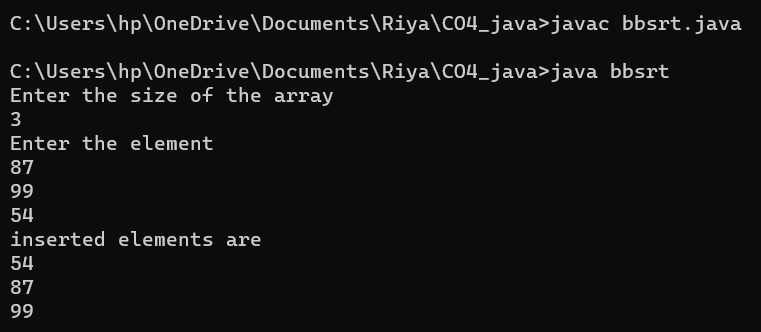
System.out.println(arr[i]);

}

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 24**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.ArrayList;

import java.util.Comparator;

import java.util.Scanner;

import java.util.Collections;

public class ArrayList Demo{

public static void insert(ArrayList<String> list){

Scanner Snr = new Scanner(System.in);

System.out.print("Enter the string to be added: ");

String val = Snr.next();

list.add(val);

System.out.println("Value inserted successfully");

}

public static void deleteAll(ArrayList<String> list){

list.clear();

System.out.println("ArrayList successfully cleared");

}

public static void find(ArrayList<String> list){

Scanner Snr = new Scanner(System.in);

System.out.print("Enter the index to get: ");

int val = Snr.nextInt();

System.out.println(list.get(val));

}

public static void delete(ArrayList<String> list){

Scanner Snr = new Scanner(System.in);

System.out.print("Enter the index to delete: ");

int val = Snr.nextInt();

list.remove(val);

System.out.println("Value successfully removed");

}

public static void len(ArrayList<String> list){

System.out.println("Length of Array List: " + list.size());

}

public static void sortlist(ArrayList<String> list){

list.sort(Comparator.naturalOrder());

System.out.println("Sorted Array List: " + list);

}

public static void sortRev(ArrayList<String> list){

list.sort(Comparator.reverseOrder());

System.out.println("Reverse Order Sorted Array List: " + list);

}

public static void max(ArrayList<String> list){

String large = list.get(0);

int len = large.length();

for(int i=1; i<list.size(); i++){

String cand = list.get(i);

if(len < cand.length()){

large = list.get(i);

len = large.length();

}

}

System.out.println("Largest String: " + large);

}

public static void display(ArrayList<String> list){

System.out.println(list);

}

public static void main(String[] args){

ArrayList<String> list = new ArrayList<String>();

Scanner Snr = new Scanner(System.in);

int choice=0;

while(choice!=10){

System.out.println("");

System.out.println(":: ARRAYLIST ::");

System.out.println("1. Insert String");

System.out.println("2. Clear List");

System.out.println("3. Get Specific Element by Index");

System.out.println("4. Remove Specific Element by Index");

System.out.println("5. Length of List");

System.out.println("6. Sort Array List");

System.out.println("7. Reverse Order Sort of Array List");

System.out.println("8. Get Longest value from List");

System.out.println("9. Display");

System.out.println("10. Exit");

System.out.print("Enter your choice: ");

choice= Snr.nextInt();

switch(choice){

case 1:

insert(list);

break;

case 2:

deleteAll(list);

break;

case 3:

find(list);

break;

case 4:

delete(list);

break;

case 5:

len(list);

break;

case 6:

sortlist(list);

break;

case 7:

sortRev(list);

break;

case 8:

max(list);

break;

case 9:

display(list);

break;

case 10:

System.exit(0);

break;

default:

System.out.println("Enter a valid choice!");

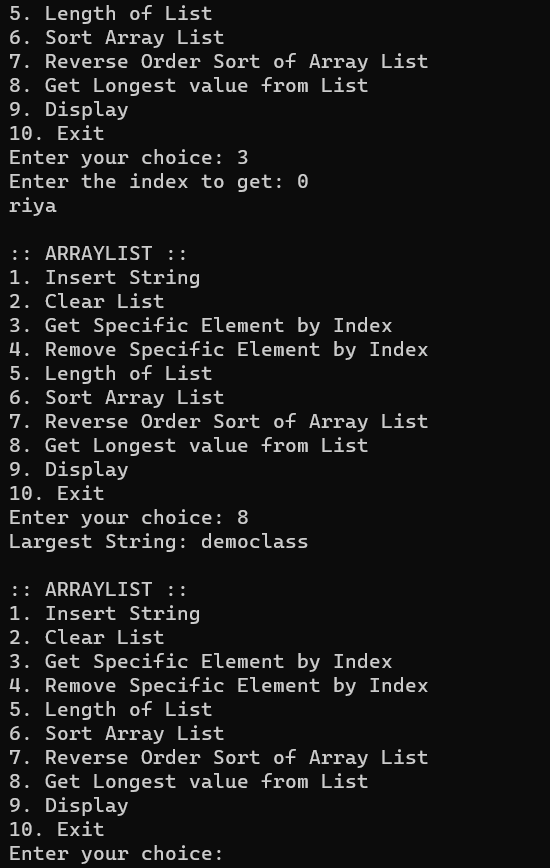
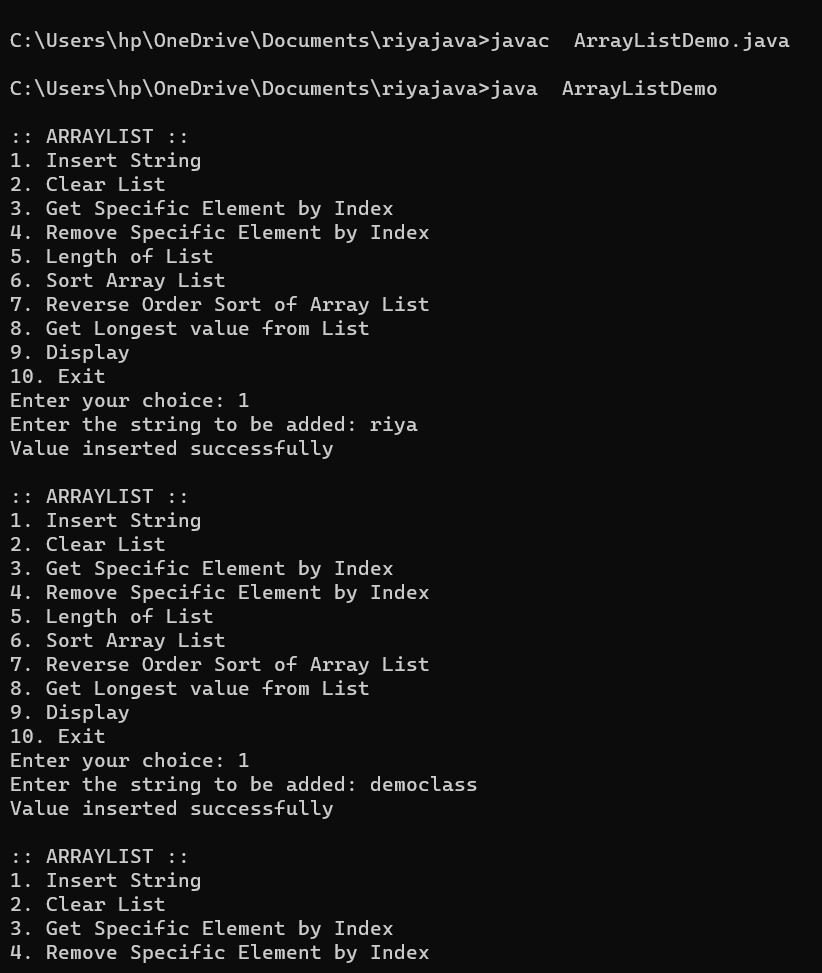
}

}

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 25**

**Aim:**

Program to remove all the elements from a linked list.

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.LinkedList;

import java.util.\*;

public class LinkedListDemo{

public static void main(String[] args){

LinkedList<String> list = new LinkedList<String>();

Scanner sc = new Scanner(System.in);

System.out.println("Enter the no of linked list: ");

int n=sc.nextInt();

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

System.out.println("Enter the items to the linked list");

String item=sc.next();

list.add(item);

}

System.out.println("Original LinkedList: "+list);

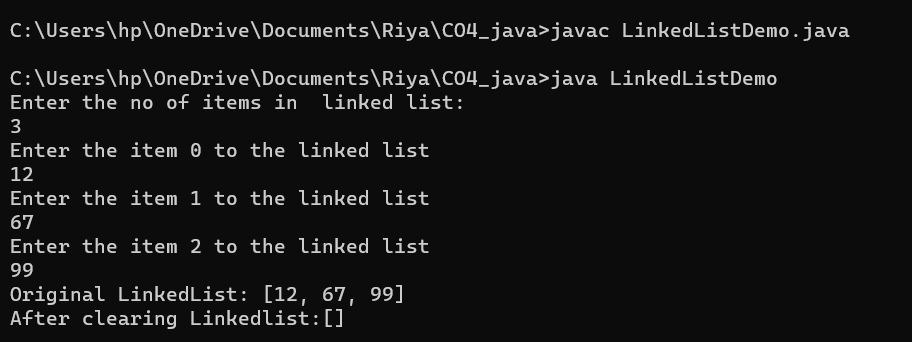
list.clear();

System.out.println("After clearing Linkedlist:" +list);

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 26**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.\*;

public class StackDemo{

public static void main(String[] args){

Stack<String> stack = new Stack<String>();

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of elemnts of Stack : ");

int n=sc.nextInt();

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

System.out.println("Enter the items");

String list=sc.next();

stack.add(list);

}

System.out.println("The items of the stack are: ");

System.out.println(stack);

System.out.println("Enter the position to be removed: ");

int remove=sc.nextInt();

//removing the elements

String rem\_ele=stack.remove(remove);

System.out.println("The final Stack is: ");

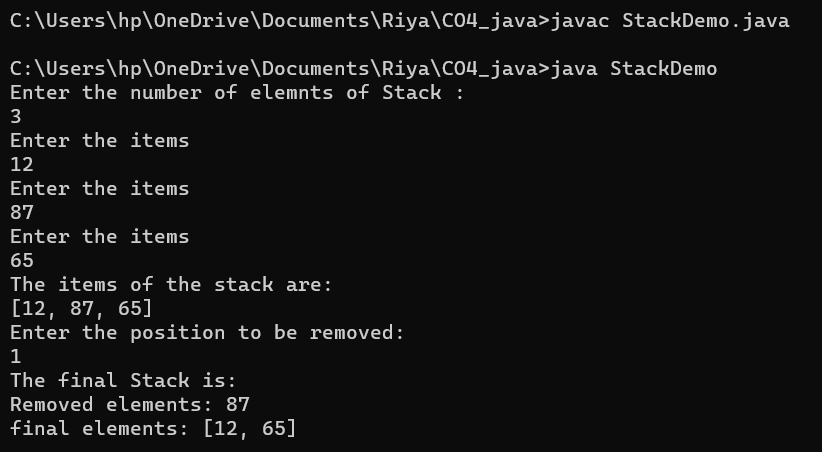
System.out.println("Removed elements: "+rem\_ele);

System.out.println("final elements: "+stack);

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 27**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.\*;

public class PriorityQueueDemo{

public static void main(String[] args){

PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>();

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of elemnts of Queue: ");

int n=sc.nextInt();

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

System.out.println("Enter the items: ");

int integer=sc.nextInt();

pQueue.add(integer);

}

System.out.println("The Queue is: ");

System.out.println(pQueue);

    System.out.println("The top element of Queue is: ");

System.out.println(pQueue.peek());

//System.out.println(pQueue.poll());

System.out.println(" remove from the top of Queue");

System.out.println(pQueue.poll());

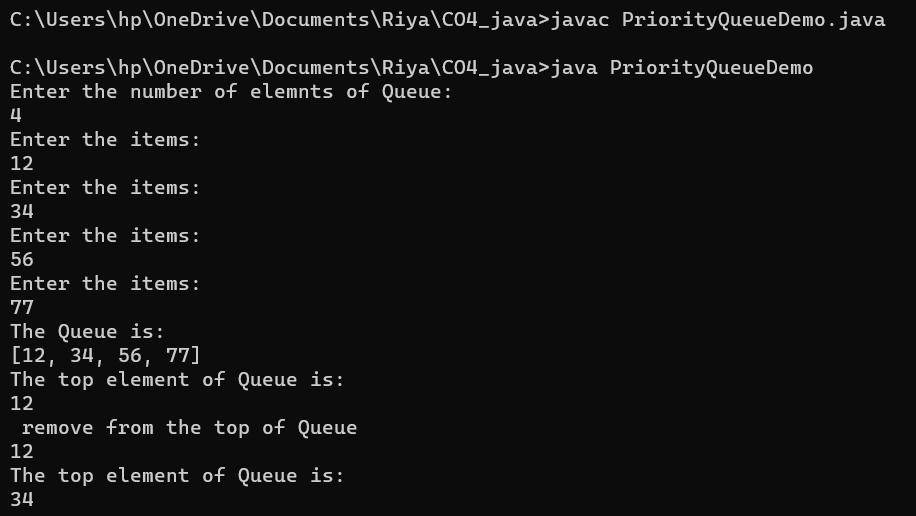
System.out.println("The top element of Queue is: ");

System.out.println(pQueue.peek());

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 28**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.ArrayDeque;

import java.util.Deque;

public class AandD{

public static void main(String[] args){

Deque<Integer> deque=new ArrayDeque<>();

deque.addFirst(1);

deque.addLast(2);

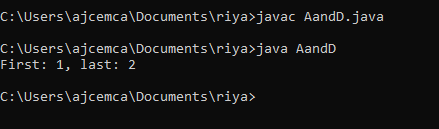
int first=deque.removeFirst();

int last=deque.removeLast();

System.out.println("First: "+first+", last: "+last);

}}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 29**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.\*;

public class GFG1{

public static void main(String[] args){

LinkedHashSet<Integer> hs1 = new LinkedHashSet<Integer>();

Scanner sc = new Scanner(System.in);

System.out.println("Enter the no of set elements: ");

int n=sc.nextInt();

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

System.out.println("Enter the items to the SET 1");

int item=sc.nextInt();

hs1.add(item);

}

System.out.println("The Set One  is: ");

System.out.println(hs1);

System.out.println(" ");

LinkedHashSet<Integer> hs2 = new LinkedHashSet<Integer>();

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

System.out.println("Enter the items to the SET 2");

int item1=sc.nextInt();

hs2.add(item1);

}

System.out.println("The Set Two is: ");

System.out.println(hs2);

System.out.println(" ");

//Intersection OPeration

LinkedHashSet<Integer> intersection = new LinkedHashSet<Integer>(hs1);

intersection.retainAll(hs2);

System.out.println("TheINTERSECTION of the sets is: "+intersection);

System.out.println(" ");

//Union OPeration

LinkedHashSet<Integer> union = new LinkedHashSet<Integer>(hs1);

union.addAll(hs2);

System.out.println("The UNION of the sets is: "+union);

System.out.println(" ");

//difference Operation

LinkedHashSet<Integer> diff = new LinkedHashSet<Integer>(hs1);

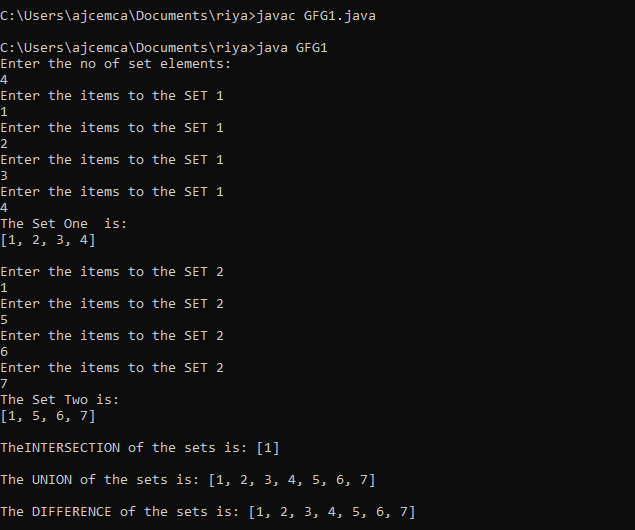
diff.removeAll(hs2);

System.out.println("The DIFFERENCE of the sets is: "+union);

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 30**

**Aim:**

Write a Java program to compare two hash set.

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.\*;

public class Main {

public static void main(String[] args) {

int n;

String str;

HashSet<String> set1= new HashSet<String>();

System.out.println("HashSet 1");

System.out.println("Enter No. of countries:");

Scanner sc=new Scanner(System.in);

n=sc.nextInt();

System.out.println("Enter the name of countries:");

Scanner sc1=new Scanner(System.in);

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

str=sc1.nextLine();

set1.add(str);

}

System.out.println("HashSet 2");

HashSet<String> set2= new HashSet<String>();

System.out.println("Enter No. of countries:");

n=sc.nextInt();

System.out.println("Enter the name of countries:");

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

str=sc1.nextLine();

set2.add(str); }

System.out.println("Set1:"+set1);

System.out.println("Set2:"+set2);

HashSet<String> a= new HashSet<String>(set1);

a.addAll(set2);

System.out.println("Union of country set:"+a);

HashSet<String> b= new HashSet<String>(set1);

b.retainAll(set2);

System.out.println("Intersection of country set:"+b);

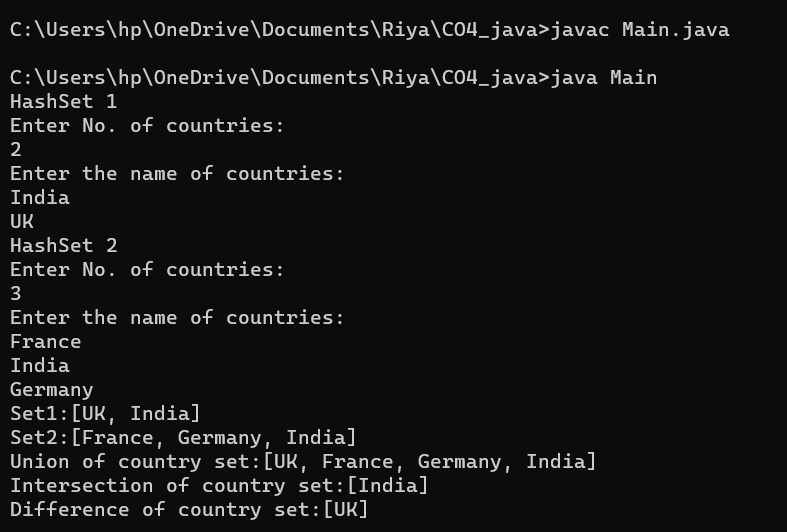
HashSet<String> c= new HashSet<String>(set1);

c.removeAll(set2);

System.out.println("Difference of country set:"+c);

}}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 31**

**Aim:**

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

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.HashMap;

public class Map {

public static void main(String[] args)

{

HashMap<String, Integer> map = new HashMap<>();

map.put("Germany", 4);

map.put("England", 1);

map.put("Brazil", 5);

System.out.println("Size of map is:- "

+ map.size());

// Printing elements in object of Map

System.out.println(map);

// Checking if a key is present and if

// present, print value by passing

// random element

if (map.containsKey("England")) {

// Mapping

Integer a = map.get("England");

// Printing value fr the corresponding key

System.out.println("value for key"

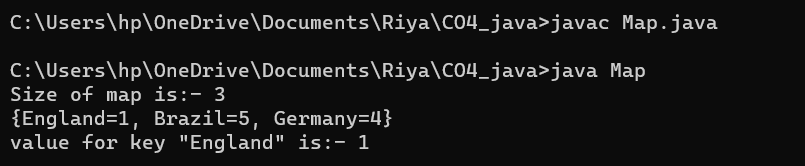
+ " \"England\" is:- " + a);

}

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 32**

**Aim:**

Program to Convert HashMap to TreeMap.

**CO4:**

Implement packages, exception handling, multithreading and generic programming by using the java.util package and Collection framework.

**Procedure:**

import java.util.\*;

public class HMap{

public static void main(String[] args) {

// TODO Auto-generated method stub

Map<Integer,String> hm=new LinkedHashMap<>();

hm.put(1,"England");

hm.put(1,"Spain");

hm.put(2,"France");

hm.put(5,"Brazil");

hm.put(2,"Argentina");

System.out.println("HashMap:"+hm);

Map<Integer,String> tm=new TreeMap<>(hm);

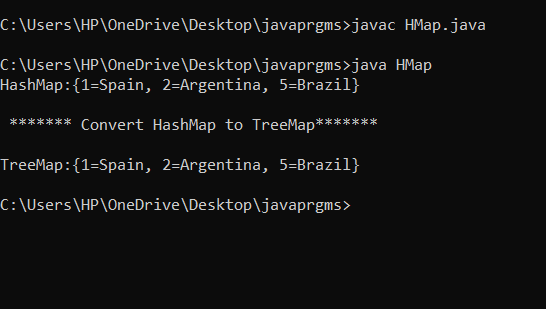
System.out.println("\n \*\*\*\*\*\*\* Convert HashMap to TreeMap\*\*\*\*\*\*\*\n");

System.out.println("TreeMap:"+tm);

}

}

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO4 was obtained.

**Experiment No.: 33**

**Aim:**

Program to draw Circle, Rectangle, Line in Applet.

**CO5:**

Design applications using files and network concepts

**Procedure:**

**applet.java**

import java.applet.\*;

import java.awt.\*;

public class applet extends Applet{

public void paint(Graphics g){

g.drawLine(20,20,200,20);

g.drawRect(20,100,200,40);

g.drawOval(20,120,200,160);

}

}

**applet.html**

<html>

<head></head>

<title>APPLET</title>

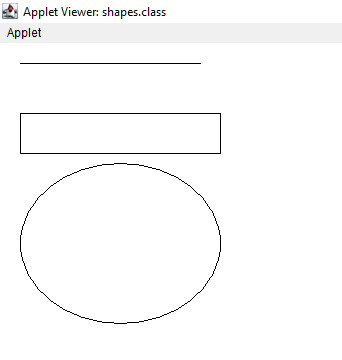
<body>

<applet code="applet.class" height="300" width="300"></applet>

</body>

</html>

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No.: 34**

**Aim:**

Program to find maximum of three numbers using AWT.

**CO5:**

Design applications using files and network concepts

**Procedure:**

import java.awt.\*;

import java.awt.Graphics;

import java.applet.\*;

import java.awt.event.\*;

public class maxThree extends Applet implements ActionListener

{

Label l1,l2,l3,l4;

TextField tf1,tf2,tf3,tf4;

Button btn;

public void init()

{

l1= new Label("Number 1:");

tf1=new TextField();

l2=new Label("Number 2:");

tf2=new TextField();

l3=new Label("Number 3:");

tf3=new TextField();

l4 = new Label("MAXIMUM:");

tf4= new TextField();

btn= new Button("Submit");

setLayout(null);

l1.setBounds(450,50,70,20);

tf1.setBounds(520,50,100,20);

l2.setBounds(450,80,70,20);

tf2.setBounds(520,80,100,20);

l3.setBounds(450,110,70,20);

tf3.setBounds(520,110,100,20);

l4.setBounds(450,140,70,20);

tf4.setBounds(520,140,100,20);

btn.setBounds(450,290,80,30);

add(l1);

add(l2);

add(l3);

add(l4);

add(tf1);

add(tf2);

add(tf3);

add(tf4);

add(btn);

btn.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==btn)

{

float n1,n2,n3,max;

n1=Float.parseFloat(tf1.getText());

n2=Float.parseFloat(tf2.getText());

n3=Float.parseFloat(tf3.getText());

if(n1>n2 && n1>n3)

{

max=n1;

}

else if(n2>n1 && n2>n3)

{

max=n2;

}

else

{

max=n3;

}

tf4.setText(String.valueOf(max));

}

}

}

**maxThree.html**

<html>

<head>

<title>Maximum of Three</title>

</head>

<body>

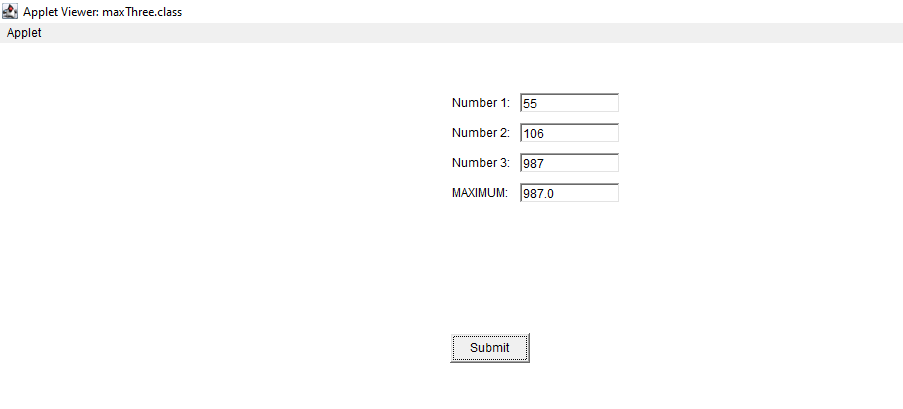
<applet code="maxThree.class" width="400" height="400">

</applet>

</body>

</html>

**Output Screenshot:**



**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No.: 35**

**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.

**CO5:**

Design applications using files and network concepts

**Procedure:**

import java.applet.\*;

import java.awt.\*;

import java.awt.Graphics;

import java.awt.event.\*;

public class Main extends Applet implements ActionListener {

Label l1,l2,l3,l4,l5,l6;

TextField t1,t2,t3,t4,t5,t6;

Button b;

public void init(){

l1 = new Label("MARK 1:");

t1 = new TextField();

l2 = new Label("MARK 2:");

t2 = new TextField();

l3 = new Label("MARK 3:");

t3 = new TextField();

l4 = new Label("MARK 4:");

t4 = new TextField();

l5 = new Label("MARK 5:");

t5 = new TextField();

l6 = new Label("PERCENTAGE:");

t6 = new TextField();

b = new Button("SEE STATUS");

setLayout(null);

l1.setBounds(450,50,70,20);

t1.setBounds(520,50,100,20);

l2.setBounds(450,80,70,20);

t2.setBounds(520,80,100,20);

l3.setBounds(450,110,70,20);

t3.setBounds(520,110,100,20);

l4.setBounds(450,140,70,20);

t4.setBounds(520,140,100,20);

l5.setBounds(450,170,70,20);

t5.setBounds(520,170,100,20);

l6.setBounds(450,200,100,20);

t6.setBounds(550,200,100,20);

b.setBounds(450,290,80,30);

add(l1);

add(l2);

add(l3);

add(l4);

add(l5);

add(l6);

add(t1);

add(t2);

add(t3);

add(t4);

add(t5);

add(t6);

add(b);

b.addActionListener(this);

}

public void actionPerformed(ActionEvent e){

float m1, m2,m3, m4,m5,percent;

m1= Float.parseFloat(t1.getText());

m2= Float.parseFloat(t2.getText());

m3= Float.parseFloat(t3.getText());

m4= Float.parseFloat(t4.getText());

m5= Float.parseFloat(t5.getText());

percent=((m1+m2+m3+m4+m5)\*100)/500;

t6.setText(String.valueOf(percent));

repaint();

}

public void paint(Graphics g){

float p;

p= Float.parseFloat(t6.getText());

if(p> 50.0) {

g.setColor(Color.YELLOW);

g.fillOval(0,0,100,100);

g.setColor(Color.black);

g.fillOval(25,25,10,10);

g.fillOval(65,25,10,10);

g.setColor(Color.black);

g.fillArc (25,35,50,50,0,-180);

}

else {

g.setColor(Color.YELLOW);

g.fillOval(0,0,100,100);

g.setColor(Color.black);

g.fillOval(25,25,10,10);

g.fillOval(75,25,10,10);

g.setColor(Color.black);

g.drawArc(25,35,50,50,0,180);

}

}

}

**#main.html**

<html>

<head></head>

<title>APPLET</title>

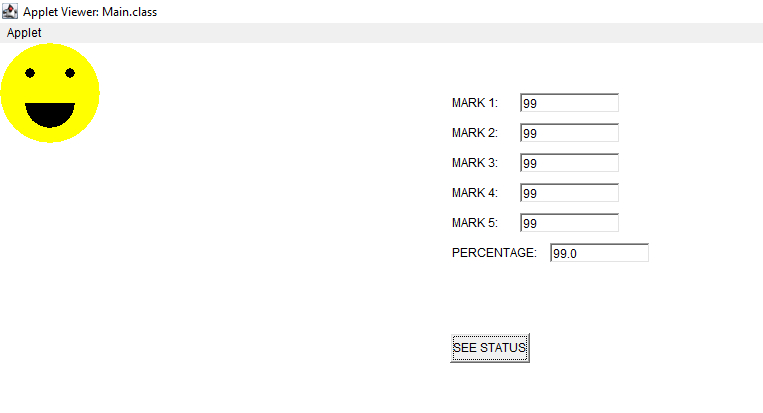
<body>

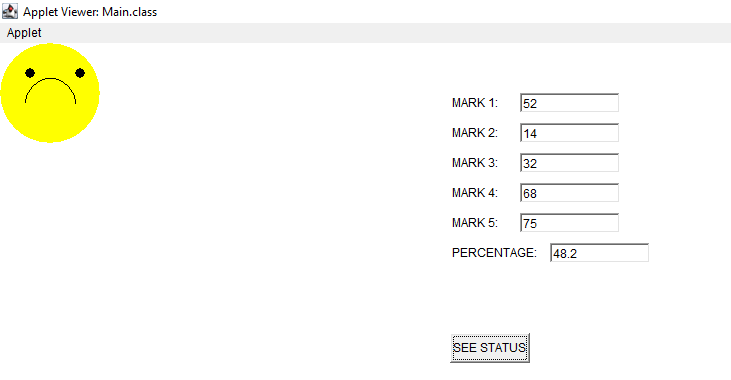
<applet code="Main.class" height="300" width="300"></applet>

</body>

</html>

**Output Screenshot:**





**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No.: 36**

**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.

**CO5:**

Design applications using files and network concepts.

**Procedure:**

**HouseApplet.java**

import java.awt.\*;

import java.awt.event.\*;

import java.applet;

public class HouseApplet extends Applet {

private Color doorColor;

public void init() {

// Set the initial color of the door

doorColor = Color.BLUE;

// Add mouse click listener

addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

// Change the color of the door to red on mouse click

doorColor = Color.RED;

repaint();

}

});

}

public void paint(Graphics g) {

// Set the background color

setBackground(Color.WHITE);

// Draw the house

g.setColor(Color.GRAY);

g.fillRect(100, 100, 200, 200); // House body

g.setColor(Color.RED);

g.fillRect(155, 200, 90, 100); // Door

g.setColor(Color.BLACK);

g.drawRect(155, 200, 90, 100); // Door outline

g.setColor(doorColor); // Set the color of the door

g.fillOval(180, 250, 10, 10); // Doorknob

g.setColor(Color.LIGHT\_GRAY);

g.fillRect(120, 150, 60, 60); // Window 1

g.fillRect(220, 150, 60, 60); // Window 2

g.setColor(Color.BLACK);

g.drawRect(120, 150, 60, 60); // Window 1 outline

g.drawRect(220, 150, 60, 60); // Window 2 outline

g.drawLine(190, 100, 100, 200); // Roof left line

g.drawLine(190, 100, 280, 200); // Roof right line

}

}

**Index.html**

<html>

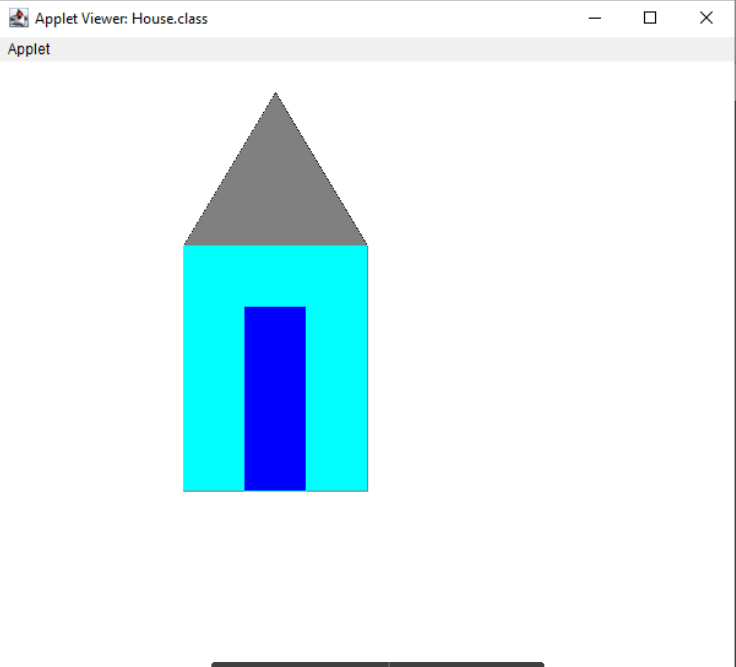
<body>

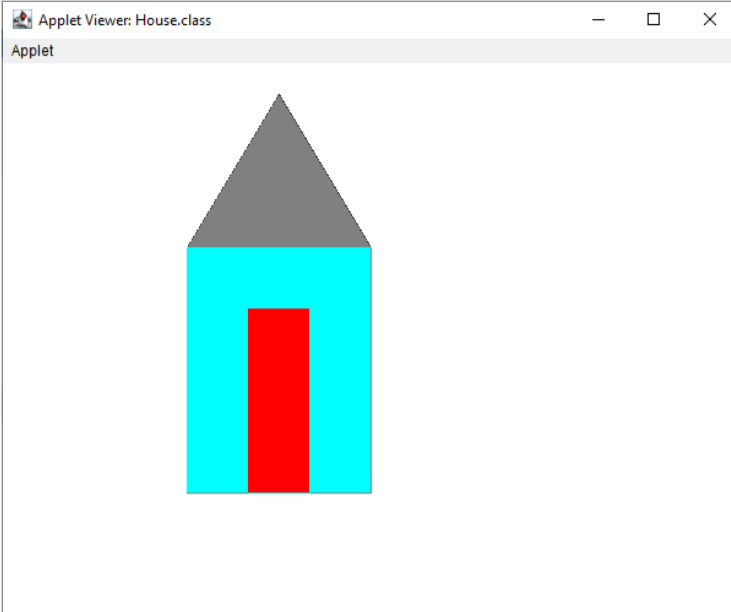
<applet code="HouseApplet.class" width="600" height="600"></applet>

</body>

</html>

**Output Screenshot:**





**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No.: 37**

**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.

**CO5:**

Design applications using files and network concepts.

**Procedure:**

import java.applet.\*;

import java.awt.\*;

import java.awt.Graphics.\*;

import java.awt.event.\*;

public class Choiceq extends Applet implements ItemListener{

int select;

Choice ch = new Choice();

public void init(){

Label l1 = new Label("Select Choice");

l1.setBounds(750,50,200,50);

add(l1);

//Choice ch = new Choice();

ch.addItem("Select Your Choice");

ch.addItem("Rectangle");

ch.addItem("Square");

ch.addItem("Circle");

ch.addItem("Triangle");

add(ch);

ch.addItemListener(this);

}

public void itemStateChanged(ItemEvent e){

select = ch.getSelectedIndex();

repaint();

}

public void paint(Graphics g){

g.setColor(Color.blue);

super.paint(g);

if(select ==1){

g.drawRect(280,100,160,40);

}

if(select == 2){

g.drawRect(50,50,100,100);

}

if(select == 3){

g.drawOval(150,150,100,100);

}

if(select == 4){

g.drawLine(120,130,280,130);

g.drawLine(120,130,200,65);

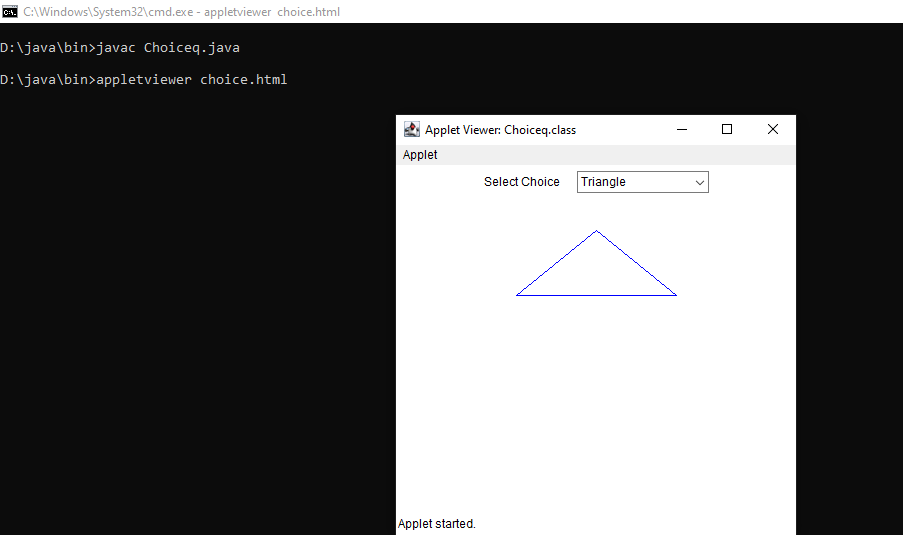
g.drawLine(200,65,280,130);

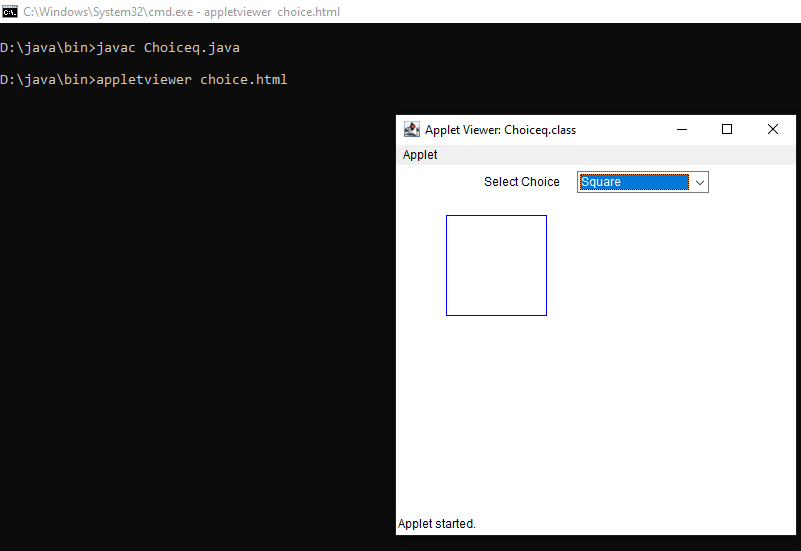
}

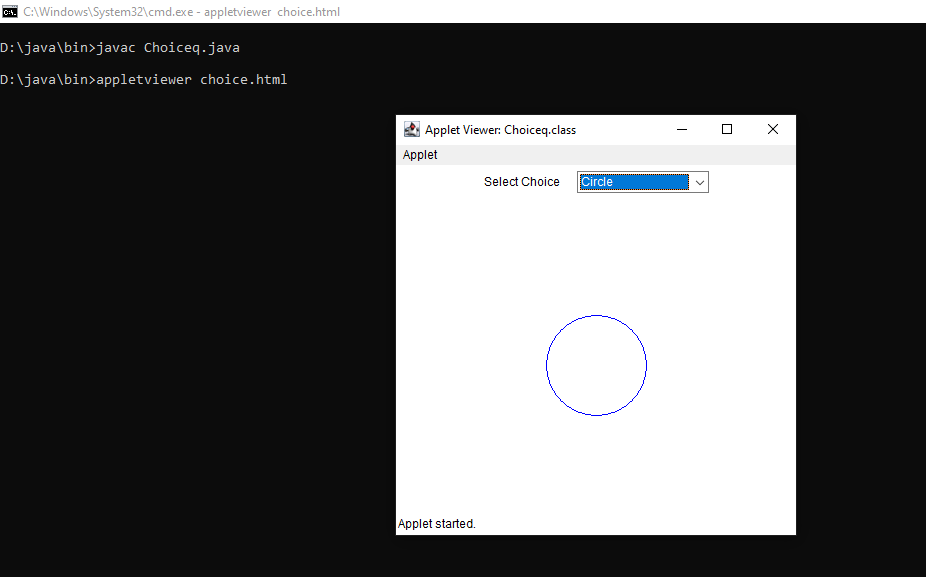
}

}

**Output Screenshot:**







**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No.: 38**

**Aim:**

Develop a program to handle all mouse events and window events.

**CO5:**

Design applications using files and network concepts

**Procedure:**

import java.awt.\*;

import java.awt.event.\*;

public class MEvt\_WEvt extends Frame implements MouseMotionListener, MouseListener

{

static Label label1, label2, label3, label4, label5;

public MEvt\_WEvt()

{

setTitle("All Mouse Events and Window Events");

setSize(900, 300);

setLayout(new FlowLayout());

addMouseMotionListener(this);

addMouseListener(this);

Panel p = new Panel();

Panel p1 = new Panel();

Label l1 = new Label("MouseMotionListener events:");

Label l2 = new Label("MouseListener events:");

label1 = new Label("no event");

label2 = new Label("no event");

label3 = new Label("no event");

label4 = new Label("no event");

label5 = new Label("no event");

p.add(l1);

p.add(label1);

p.add(label2);

p1.add(l2);

p1.add(label3);

p1.add(label4);

p1.add(label5);

add(p);

add(p1);

addWindowListener(new WindowAdapter()

{

public void windowClosing(WindowEvent e)

{

System.exit(0);

}

});

setVisible(true);

}

public static void main(String[] args)

{

new MEvt\_WEvt();

}

public void mouseDragged(MouseEvent e)

{

label1.setText("mouse is dragged through point " + e.getX() + " " + e.getY());

}

public void mouseMoved(MouseEvent e)

{

label2.setText("mouse is moved to point " + e.getX() + " " + e.getY());

}

public void mousePressed(MouseEvent e)

{

label3.setText("mouse pressed at point:" + e.getX() + " " + e.getY());

}

public void mouseReleased(MouseEvent e)

{

label3.setText("mouse released at point:" + e.getX() + " " + e.getY());

}

public void mouseExited(MouseEvent e)

{

label4.setText("mouse exited through point:" + e.getX() + " " + e.getY());

}

public void mouseEntered(MouseEvent e)

{

label4.setText("mouse entered at point:" + e.getX() + " " + e.getY());

}

public void mouseClicked(MouseEvent e)

{

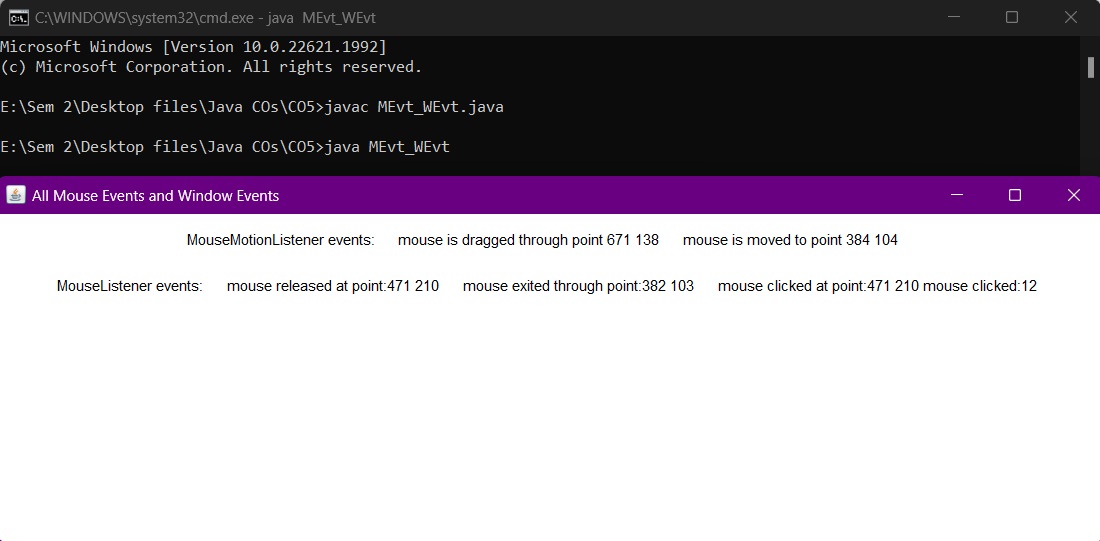
label5.setText("mouse clicked at point:" + e.getX() + " " + e.getY() + " mouse clicked:" +

e.getClickCount());

}

}

**Output Screenshot:**

****

**Result**

The program was executed and the result was successfully obtained. Thus CO5 was obtained.

**Experiment No.: 39**

**Aim:**

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

**CO6:**

Design applications using files and networking concepts.

**Procedure:**

import java.io.\*;

import java.util.Scanner;

public class WrtRead

{

public static void main(String[] args)

{

try

{

FileWriter writer = new FileWriter("output.txt");

Scanner scanner = new Scanner(System.in);

System.out.println("Enter lines of text (type 'done' on a separate line to finish):");

String inputLine;

while (!(inputLine = scanner.nextLine()).equalsIgnoreCase("done"))

{

writer.write(inputLine + "\n");

}

writer.close();

System.out.println("Data written to the file successfully.");

FileReader reader = new FileReader("output.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

String line;

System.out.println("Contents of the file:");

while ((line = bufferedReader.readLine()) != null)

{

System.out.println(line);

}

bufferedReader.close();

scanner.close();

}

catch (IOException e)

{

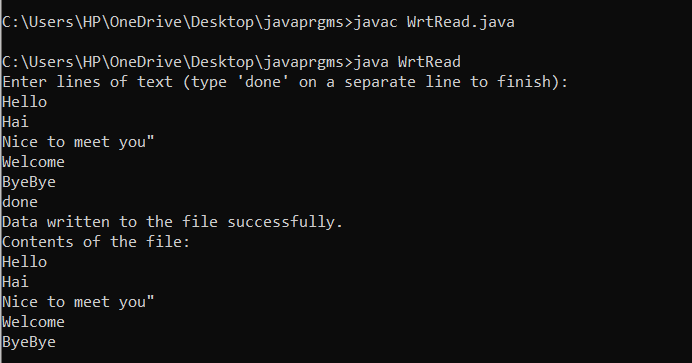
e.printStackTrace();

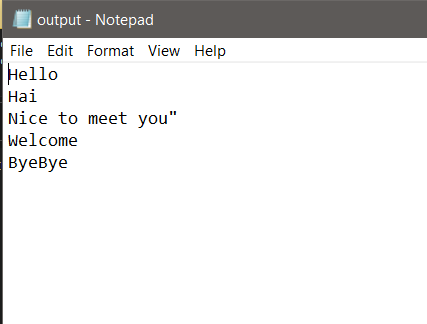
}

}

}

**Output Screenshot:**





**Result**

The program was executed and the result was successfully obtained. Thus CO6 was obtained.

**Experiment No.: 40**

**Aim:**

Write a program to copy one file to another.

**CO6:**

Design applications using files and networking concepts.

**Procedure:**

import java.io.\*;

public class Pgm45

{

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

{

FileInputStream fileinput=new FileInputStream("abc.txt") ;

FileOutputStream fileoutput=new FileOutputStream("2.txt");

int i;

while ((i=fileinput.read()) !=-1)

{

fileoutput.write(i);

}

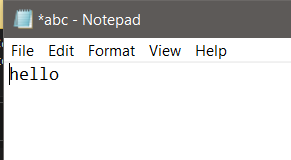
System.out.println("successfully copied");

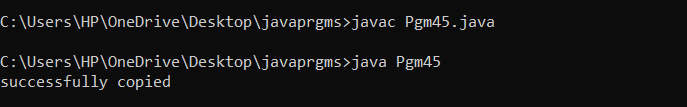
}

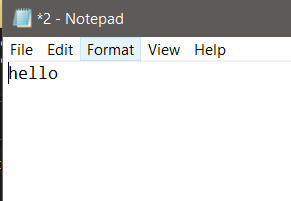
}

}

**Output Screenshot:**

****

****

****

**Result**

The program was executed and the result was successfully obtained. Thus CO6 was obtained.

**Experiment No.: 41**

**Aim:**

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

**CO6:**

Design applications using files and networking concepts.

**Procedure:**

import java.io.\*;

public class OddEven {

public static void main(String[] args) {

try {

FileReader reader = new FileReader("numbers.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

FileWriter evenWriter = new FileWriter("evens.txt");

FileWriter oddWriter = new FileWriter("odds.txt");

String line;

while ((line = bufferedReader.readLine()) != null) {

int number = Integer.parseInt(line);

if (number % 2 == 0){

evenWriter.write(number + "\n");

} else {

oddWriter.write(number + "\n");

}

}

bufferedReader.close();

evenWriter.close();

oddWriter.close();

System.out.println("Even and odd numbers separated and copied successfully.");

}

catch (IOException | NumberFormatException e)

{

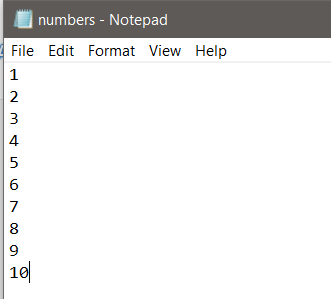
e.printStackTrace();

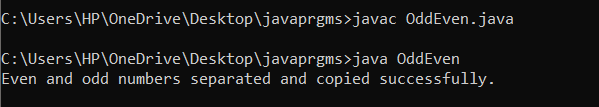
}

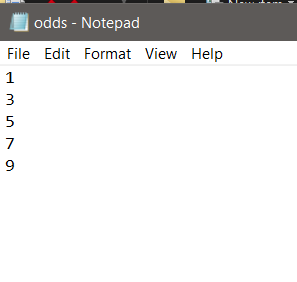
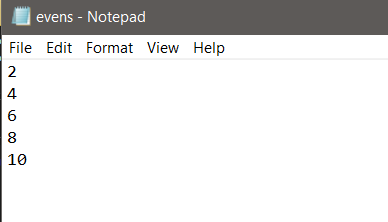
}

}

**Output Screenshot:**

****

****

****

**Result**

The program was executed and the result was successfully obtained. Thus CO6 was obtained.

**Experiment No.: 42**

**Aim:**

Client Server communication using DatagramSocket - UDP

**CO6:**

Design applications using files and networking concepts.

**Procedure:**

**Client.java**

import java.net.DatagramPacket;

import java.net.DatagramSocket;

import java.net.InetAddress;

import java.util.Scanner;

public class Client {

public static void main(String[] args) {

try {

DatagramSocket socket = new DatagramSocket();

InetAddress serverAddress = InetAddress.getByName("localhost");

int serverPort = 9876;

Scanner scanner = new Scanner(System.in);

while (true) {

byte[] sendData;

System.out.print("Client: ");

String clientMessage = scanner.nextLine();

sendData = clientMessage.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);

socket.send(sendPacket);

if (clientMessage.equals("done")) {

System.out.println("Client is exiting...");

break;

}

byte[] receiveData = new byte[1024];

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

socket.receive(receivePacket);

String serverMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());

System.out.println("Server: " + serverMessage);

if (serverMessage.equals("done")){

System.out.println("Server is exiting...");

break;

}

}

socket.close();

scanner.close();

}

catch (Exception e){

e.printStackTrace();

}

}

}

**Server.java**

import java.net.DatagramPacket;

import java.net.DatagramSocket;

public class Server {

public static void main(String[] args){

try {

DatagramSocket socket = new DatagramSocket(9876);

byte[] receiveData = new byte[1024];

byte[] sendData;

while (true) {

DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

socket.receive(receivePacket);

String clientMessage = new String(receivePacket.getData(), 0, receivePacket.getLength());

System.out.println("Client: " + clientMessage);

if (clientMessage.equals("done")) {

System.out.println("Server is exiting...");

break;

}

System.out.print("Server: ");

String serverMessage = System.console().readLine();

sendData = serverMessage.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, receivePacket.getAddress(), receivePacket.getPort());

socket.send(sendPacket);

if (serverMessage.equals("done")) {

System.out.println("Server is exiting...");

break;

}

}

socket.close();

}

catch (Exception e)

{

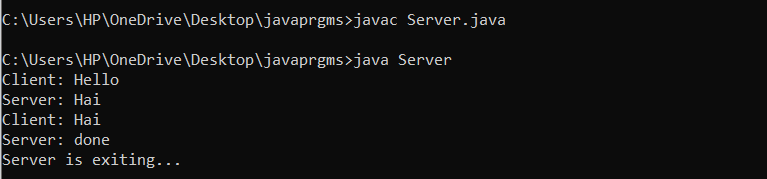
e.printStackTrace();

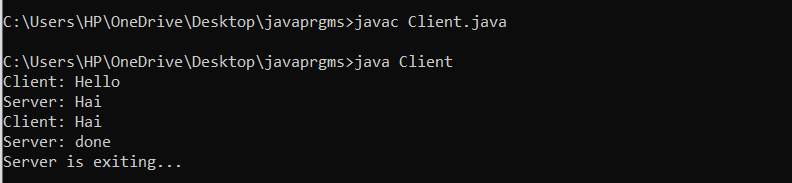
}

}

}

**Output Screenshot:**

****

****

**Result**

The program was executed and the result was successfully obtained. Thus CO6 was obtained.