**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.\*;

class product {

    Scanner sc=new Scanner (System.in);

    String pcode,pname;

    int price;

    void getfn(){

        System.out.print("\nEnter the pcode :");

        pcode=sc.nextLine();

        System.out.print("Enter the pname :");

        pname=sc.nextLine();

        System.out.print("Enter the price :");

        price=sc.nextInt();

    }

    void put(){

        System.out.print("\nCode of the product  :"+pcode);

        System.out.print("\nName of the product  :"+pname);

        System.out.print("\nPrice of the product :"+price);

    }

}

public class minproduct{

    public static void main(String [] args){

        Scanner sc= new Scanner(System.in);

        product p1=new product();

        product p2=new product();

        product p3=new product();

        p1.getfn();

        p2.getfn();

        p3.getfn();

        if(p1.price<p2.price){

            if(p1.price<p2.price){

                System.out.println("First product has the smallest price");

                p1.put();

            }

            else{

                System.out.println("Second product has the smallest price");

                p2.put();

            }

        }

        else{

            if(p2.price<p3.price){

                System.out.println("Second product has the smallest price");

                p2.put();

            }

            else{

                System.out.println("Third product has the smallest price");

                p3.put();

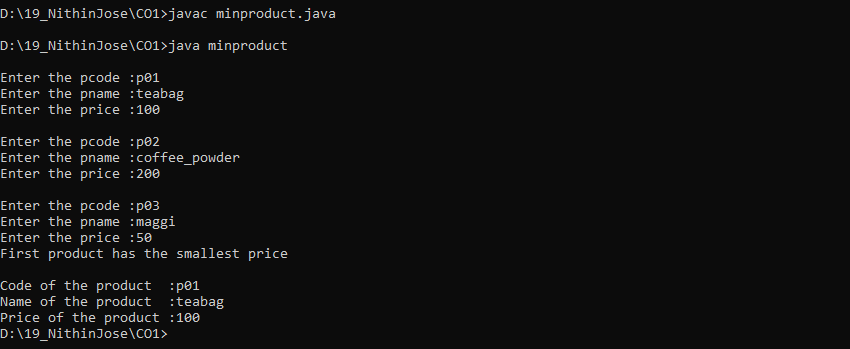
            }

        }

    }

}

**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.\*;

class MatrixClass {

    Scanner sc = new Scanner(System.in);

    public int[][] setfn(int r, int c) {

        int[][] arr = new int[r][c];

        for (int i = 0; i < r; i++) {

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

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

                arr[i][j] = sc.nextInt();

            }

        }

        return arr;

    }

}

public class Matrix {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int r1 = sc.nextInt();

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

        int c1 = sc.nextInt();

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

        int r2 = sc.nextInt();

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

        int c2 = sc.nextInt();

// ADDITION

        if ((r1 != r2) || (c1 != c2)) {

            System.out.println("Addition is not Possible");

        } else {

            System.out.println("Enter Matrix 1:");

            MatrixClass m1 = new MatrixClass();

            int[][] arr1 = m1.setfn(r1, c1);

            System.out.println("Enter Matrix 2:");

            MatrixClass m2 = new MatrixClass();

            int[][] arr2 = m2.setfn(r2, c2);

            int[][] res\_arr = new int[r1][c1];

            for (int i = 0; i < r1; i++) {

                for (int j = 0; j < c1; j++) {

                    res\_arr[i][j] = arr1[i][j] + arr2[i][j];

                }

            }

            System.out.println("Resultant Matrix:");

            for (int i = 0; i < r1; i++) {

                for (int j = 0; j < c1; j++) {

                    System.out.print(res\_arr[i][j] + " ");

                }

                System.out.println();

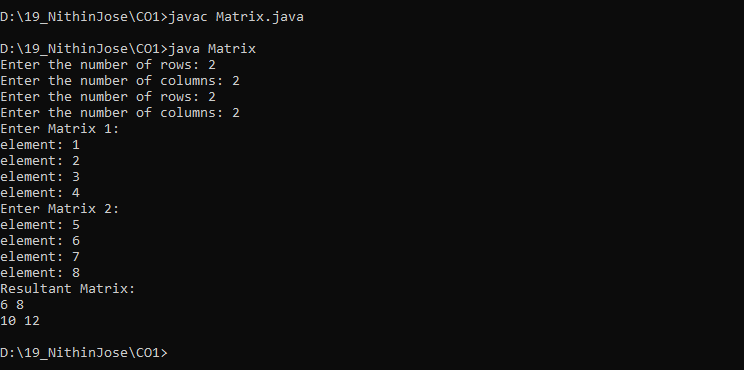
            }

        }

    }

}

**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.\*;

public class complexNumber {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("\nEnter real part of first Number :");

        int a1 = sc.nextInt();

        System.out.print("Enter the imaginary part of the first Number :");

        int b1 = sc.nextInt();

        System.out.print("\nEnter real part of second Number :");

        int a2 = sc.nextInt();

        System.out.print("Enter the imaginary part of the second Number :");

        int b2 = sc.nextInt();

        System.out.println("\n" + a1 + " +" + b1 + "i   +");

        System.out.println(a2 + " +" + b2 + "i");

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

        int a3 = a1 + a2;

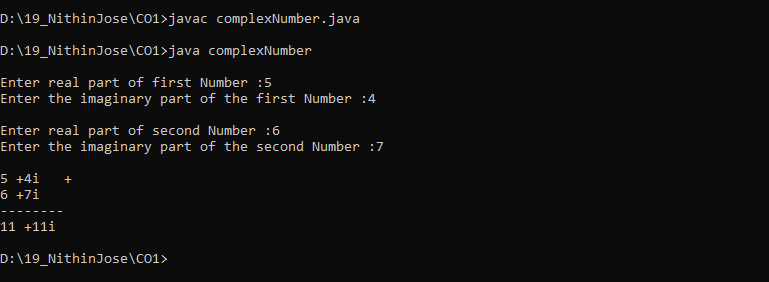
        int b3 = b1 + b2;

        System.out.println(a3 + " +" + b3 + "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 symMatrix {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int r = sc.nextInt();

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

        int c = sc.nextInt();

        int flag = 1;

        int arr1[][] = new int[r][c];

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

        for (int i = 0; i < r; i++) {

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

                arr1[i][j] = sc.nextInt();

            }

        }

        System.out.print("\nMatrix Entered");

        for (int i = 0; i < r; i++) {

            System.out.print("\n");

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

                System.out.print(arr1[i][j] + " ");

            }

        }

        // PRINTING TRANSPOSE

        System.out.print("\nMatrix Transpose");

        for (int i = 0; i < r; i++) {

            System.out.print("\n");

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

                System.out.print(arr1[j][i] + " ");

            }

        }

        // CHECKING

        for (int i = 0; i < r; i++) {

            System.out.print("\n");

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

                if ((arr1[i][j]) != (arr1[j][i])) {

                    flag = 0;

                }

            }

        }

        if (flag == 0) {

            System.out.println("MATRIX IS NOT SYMMETRIC");

        } else {

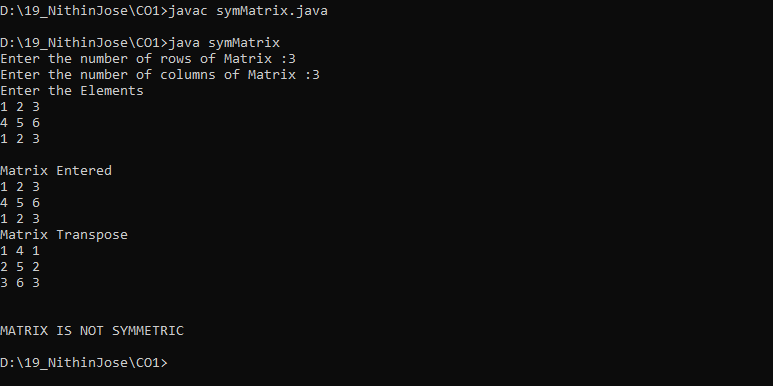
            System.out.println("MATRIX IS SYMMETRIC IN CHARACTER");

        }

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 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.

**CO1**

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

**Procedure**

import java.util.\*;

class CPU {

    int price;

    CPU() {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the price: ");

        price = sc.nextInt();

    }

    public void display() {

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

    }

    class Processor {

        int core;

        String manufacturer;

        Processor() {

            Scanner obj1 = new Scanner(System.in);

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

            core = obj1.nextInt();

            System.out.print("Enter the Manufacturer: ");

            manufacturer = obj1.next();

        }

        public void display() {

            System.out.println("Number of Cores: " + core);

            System.out.println("Manufacturer (Processor): " + manufacturer);

        }

    }

    static class RAM {

        int memory;

        String manufacturerRAM;

        RAM() {

            Scanner obj2 = new Scanner(System.in);

            System.out.print("Enter the Memory Size: ");

            memory = obj2.nextInt();

            System.out.print("Enter the Manufacturer (RAM): ");

            manufacturerRAM = obj2.next();

        }

        public void display() {

            System.out.println("Memory size of RAM: " + memory);

            System.out.println("Manufacturer of RAM: " + manufacturerRAM);

        }

    }

}

public class CpuDetails {

    public static void main(String args[]) {

        CPU c = new CPU();

        CPU.Processor s = c.new Processor();

        CPU.RAM p = new CPU.RAM();

        c.display();

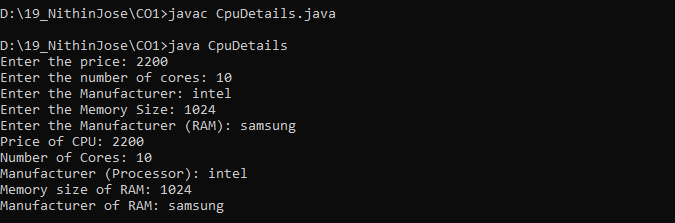
        s.display();

        p.display();

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 6**

**Aim**

Program to Sort strings

**CO2**

Implement arrays and strings.

**Procedure**

import java.util.Scanner;

public class sortString {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int l = sc.nextInt();

        String str[] = new String[l];

        int i;

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

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

            str[i] = sc.next();

        }

        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++) {

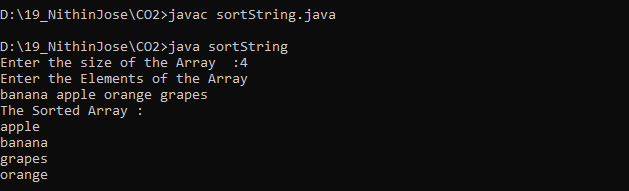
            System.out.println(str[i]);

        }

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 7**

**Aim**

Search an element in an array.

**CO2**

Implement arrays and strings.

**Procedure**

import java.util.\*;

public class StringArraySearch {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

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

        int size = scanner.nextInt();

        String[] array = new String[size];

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

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

            array[i] = scanner.next();

        }

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

        String target = scanner.next();

        boolean found = false;

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

            if (array[i].equals(target)) {

                found = true;

                break;

            }

        }

        if (found) {

            System.out.println("String found in the array.");

        } else {

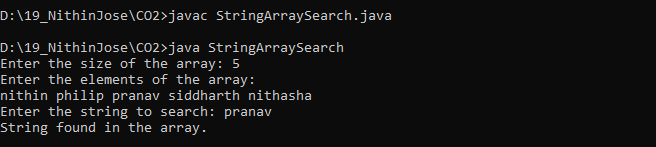
            System.out.println("String not found in the array.");

        }

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 8**

**Aim**

Perform string manipulations

**CO2**

Implement 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.print("Enter the first string: ");

        str1 = sc.nextLine();

        System.out.print("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());

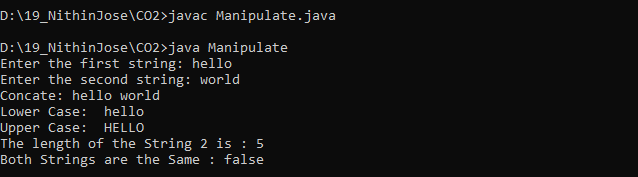
        System.out.println("The length of the String 2 is : " + str2.length());

        System.out.println("Both Strings are the Same : " + str1.equals(str2));

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 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.

**CO2**

Implement arrays and strings.

**Procedure**

import java.util.Scanner;

class Employee {

    int eNo;

    String eName;

    int eSalary;

    Scanner sc = new Scanner(System.in);

    Employee() {

        System.out.print("Enter the Employee No         :");

        eNo = sc.nextInt();

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

        eName = sc.next();

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

        eSalary = sc.nextInt();

    }

}

public class EmployeeInfo {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

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

        int n = sc.nextInt();

        Employee[] emp = new Employee[n];

        for (int i = 0; i < n; i++) {

            emp[i] = new Employee();

        }

        System.out.print("\nEnter the Employee Number :");

        int num = sc.nextInt();

        boolean flag = false;

        for (int i = 0; i < n; i++) {

            if (emp[i].eNo == num) {

                flag = true;

                break;

            }

        }

        if (flag == true) {

            System.out.print("The Employee is Found in Database");

        } else {

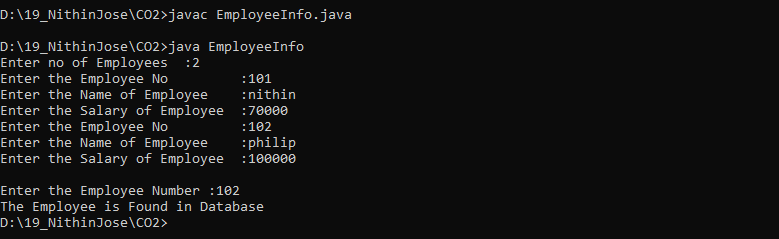
            System.out.print("The Employee is not Found in Database");

        }

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 10**

**Aim**

Area of different shapes using overloaded functions.

**CO3**

Implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure**

import java.util.Scanner;

class ShapesOverload {

    public double area(double radius) {

        return 3.14 \* radius \* radius;

    }

    public double area(double base, double height) {

        return 0.5 \* base \* height;

    }

    public double area(float side) {

        return side \* side;

    }

    public double area(double length, float breadth) {

        return length \* breadth;

    }

}

public class CO3Q1 {

    public static void main(String[] args) {

        ShapesOverload obj = new ShapesOverload();

        Scanner sc = new Scanner(System.in);

        System.out.print("\nEnter the Radius of the Circle   :");

        int rc = sc.nextInt();

        System.out.print("Enter the Base of the Trangle    :");

        int bt = sc.nextInt();

        System.out.print("Enter the Height of the Triangle :");

        int ht = sc.nextInt();

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

        int ss = sc.nextInt();

        System.out.print("Enter the Side of the Rectangle  :");

        int sr = sc.nextInt();

        System.out.print("Enter the Height of the Rectangle:");

        int hr = sc.nextInt();

        System.out.println("\nArea of Circle  : " + obj.area(rc));

        System.out.println("Area of Triangle  : " + obj.area(bt, ht));

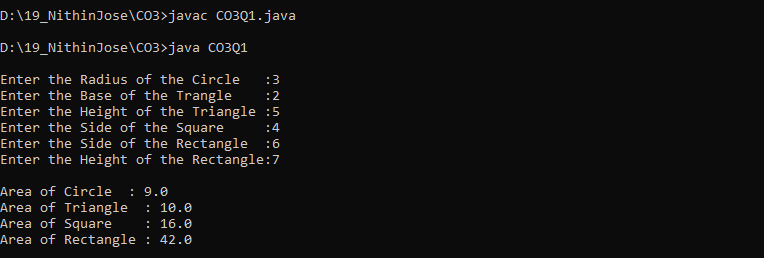
        System.out.println("Area of Square    : " + obj.area(ss));

        System.out.println("Area of Rectangle : " + obj.area(sr, hr));

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 11**

**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**

Implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure**

import java.util.Scanner;

class Employee {

    int empid;

    String name;

    int salary;

    String address;

    Employee(int a, String b, int c, String d) {

        empid = a;

        name = b;

        salary = c;

        address = d;

    }

}

class Teacher extends Employee {

    String department;

    String subject;

    Teacher(int l, String m, int n, String o, String p, String q) {

        super(l, m, n, o);

        department = p;

        subject = q;

    }

    public void display() {

        System.out.println("\nEmployee Id       :" + empid);

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

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

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

        System.out.println("Teacher Department:" + department);

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

    }

}

public class CO3Q2 {

    public static void main(String[] args) {

        int i;

        Scanner cin = new Scanner(System.in);

        System.out.print("Enter the limit of array  :");

        int n = cin.nextInt();

        Teacher e[] = new Teacher[n];

        for (i = 0; i < n; i++) {

            System.out.print("Enter the Id of the Employee  :");

            int a = cin.nextInt();

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

            String b = cin.next();

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

            int c = cin.nextInt();

            System.out.print("Enter the Address of the Employee  :");

            String d = cin.next();

            System.out.print("Enter the Department of the Teachers  :");

            String q = cin.next();

            System.out.print("Enter the Subject of the Teachers  :");

            String f = cin.next();

            e[i] = new Teacher(a, b, c, d, q, f);

        }

        System.out.println("\n\nPRINTING DETAILS");

        for (i = 0; i < n; i++) {

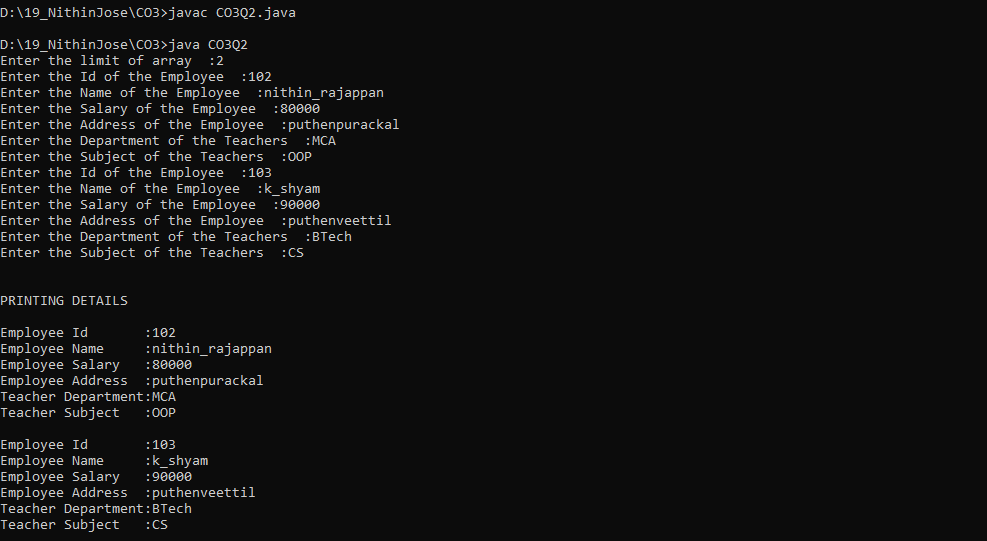
            e[i].display();

        }

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 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.

**CO3**

Implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure**

import java.util.\*;

class Person {

    String name;

    String gender;

    String address;

    int age;

    Person() {

        Scanner sc = new Scanner(System.in);

        System.out.print(" Enter name of person: ");

        name = sc.next();

        System.out.print(" Enter the gender of person: ");

        gender = sc.next();

        System.out.print(" Enter the address of person: ");

        address = sc.next();

        System.out.print(" Enter the age of person: ");

        age = sc.nextInt();

    }

}

class Employee extends Person {

    int empid;

    String companyname;

    String qualification;

    int salary;

    Employee() {

        Scanner sc = new Scanner(System.in);

        System.out.print(" Enter the empid: ");

        empid = sc.nextInt();

        System.out.print(" Enter the company name: ");

        companyname = sc.next();

        System.out.print(" Enter the qualification of employee: ");

        qualification = sc.next();

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

        salary = sc.nextInt();

    }

}

class Teacher extends Employee {

    String subject;

    String department;

    int teacherid;

    Teacher() {

        Scanner sc = new Scanner(System.in);

        System.out.print(" Enter the subject of teacher: ");

        subject = sc.nextLine();

        System.out.print(" Enter the department of teacher: ");

        department = sc.nextLine();

        System.out.print(" Enter the teacher id: ");

        teacherid = sc.nextInt();

        System.out.print("\n");

    }

    void put() {

        System.out.println("Person name is: " + name);

        System.out.println("person gender is: " + gender);

        System.out.println("person address is: " + address);

        System.out.println("person age is: " + age);

        System.out.println("employee id is: " + empid);

        System.out.println("employee company is: " + companyname);

        System.out.println("employee qualification is: " + qualification);

        System.out.println("employee salary is: " + salary);

        System.out.println("teacher subject is: " + subject);

        System.out.println("teacher department is: " + department);

        System.out.println("teacher id is: " + teacherid + "\n");

    }

}

public class CO3Q3 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        int i, n;

        System.out.print("Enter no. of records you want to insert: ");

        n = sc.nextInt();

        Teacher obj[] = new Teacher[n];

        System.out.println("Enter the details of " + n + " records\n");

        for (i = 0; i < n; i++) {

            obj[i] = new Teacher();

        }

        for (i = 0; i < n; i++) {

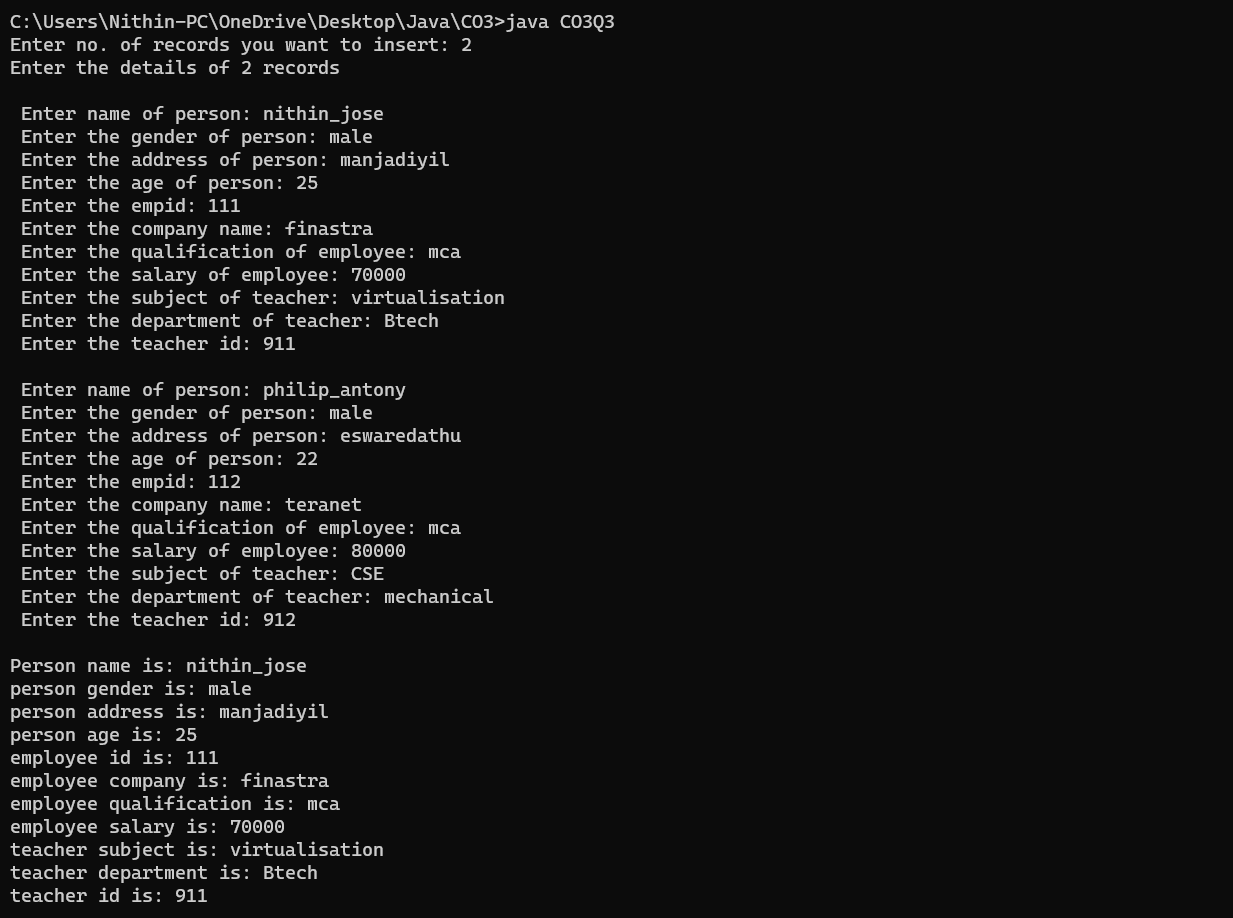
            obj[i].put();

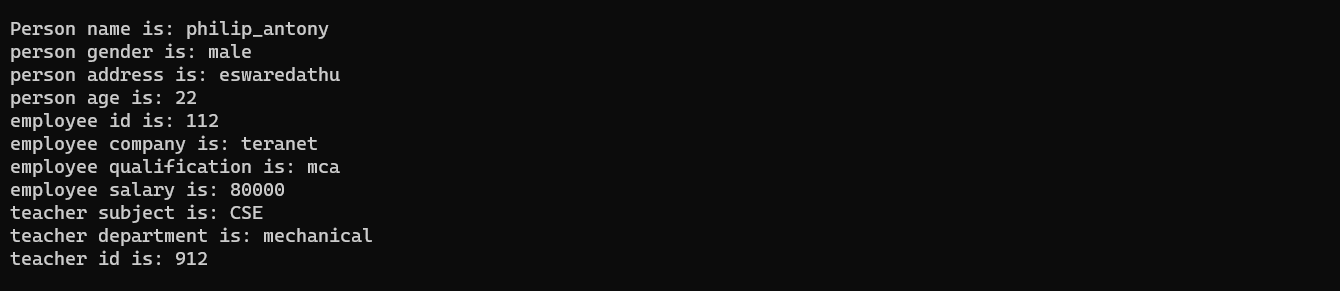
        }

    }

}

**Output Screenshot**





**Result**

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

**Experiment No.: 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.

**CO3**

Implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure**

import java.util.\*;

class Publisher {

    Scanner sc = new Scanner(System.in);

    String publisher;

    Publisher() {

        System.out.print("\nEnter the Publisher name: ");

        publisher = sc.next();

    }

}

class Book extends Publisher {

    String title;

    Book() {

        System.out.print("Enter the Title: ");

        title = sc.next();

    }

}

class Literature extends Book {

    String category;

    Literature() {

        System.out.print("Enter the category: ");

        category = sc.next();

    }

}

class Fiction extends Book {

    String category;

    Fiction() {

        System.out.print("Enter the category: ");

        category = sc.next();

    }

}

public class CO3Q4 {

    public static void main(String[] args) {

        System.out.print("\nEnter the Details of Literature Book : ");

        Literature l = new Literature();

        System.out.print("\nEnter the Details of Fiction Book    : ");

        Fiction f = new Fiction();

        System.out.println("\nPrinting Details of the book 1");

        System.out.println("Publisher: " + l.publisher);

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

        System.out.println("Category: " + l.category);

        System.out.println("\nPrinting Details of the book 2");

        System.out.println("Publisher: " + f.publisher);

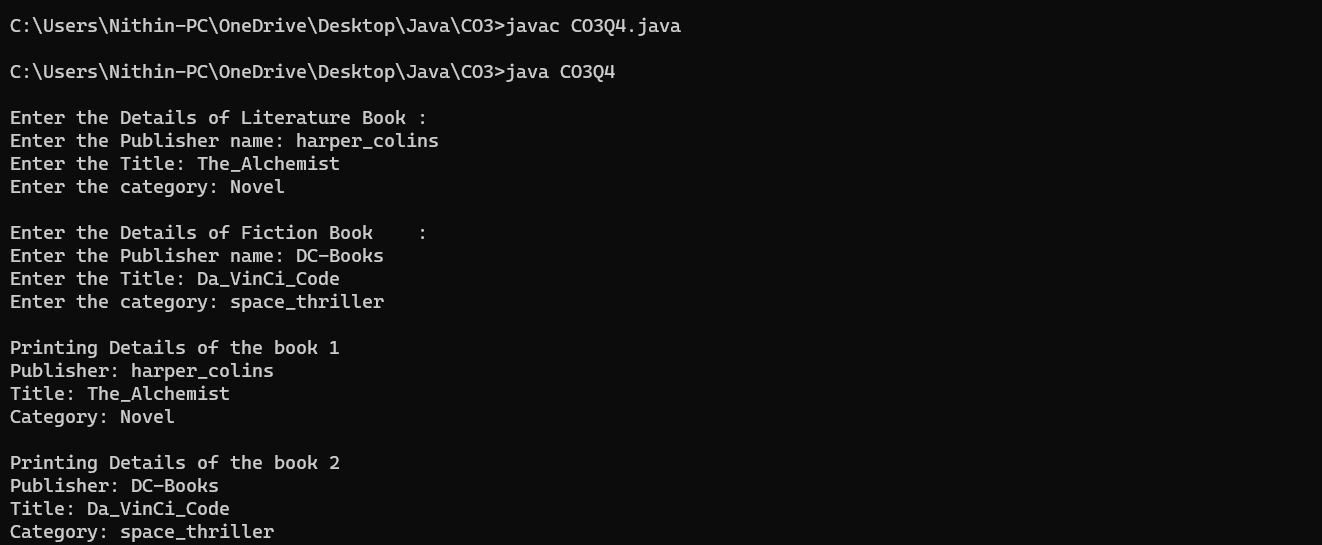
        System.out.println("Title: " + f.title);

        System.out.println("Category: " + f.category);

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 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.

**CO3**

Implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure**

import java.util.Scanner;

interface Student {

    void getAcademicMarks();

}

interface Sports {

    void getSportsMarks();

}

class Result implements Student, Sports {

    private int academicMarks;

    private int sportsMarks;

    public void getAcademicMarks() {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the marks obtained for the examination: ");

        academicMarks = sc.nextInt();

    }

    public void getSportsMarks() {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the marks/position obtained for Sports: ");

        sportsMarks = sc.nextInt();

    }

    public void displayScores() {

        System.out.println("\nAcademic Score: " + academicMarks);

        System.out.println("Sports Score: " + sportsMarks);

    }

}

public class CO3Q5 {

    public static void main(String[] args) {

        Result res = new Result();

        res.getAcademicMarks();

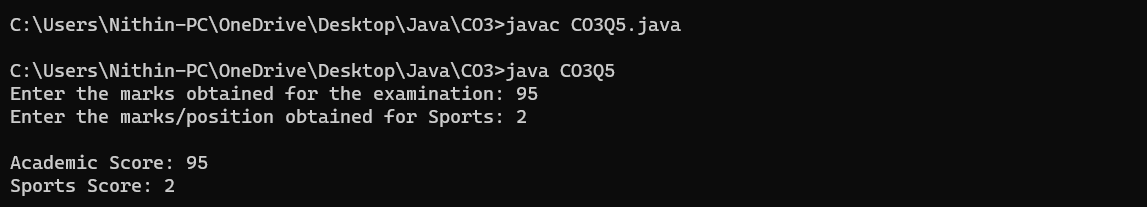
        res.getSportsMarks();

        res.displayScores();

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 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.

**CO3**

Implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure**

import java.util.Scanner;

interface Prototype {

    float area();

    float perimeter();

}

class Circle implements Prototype {

    Scanner sc = new Scanner(System.in);

    public float area() {

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

        int rad = sc.nextInt();

        float area = 3.14F \* rad \* rad;

        return area;

    }

    public float perimeter() {

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

        int rad = sc.nextInt();

        float peri = 2 \* 3.14F \* rad;

        return peri;

    }

}

class Rectangle implements Prototype {

    Scanner sc = new Scanner(System.in);

    public float area() {

        System.out.print("Enter the base of the Rectangle: ");

        int base = sc.nextInt();

        System.out.print("Enter the Height of the Rectangle: ");

        int height = sc.nextInt();

        float area = base \* height;

        return area;

    }

    public float perimeter() {

        System.out.print("Enter the base of the Rectangle: ");

        int base = sc.nextInt();

        System.out.print("Enter the Height of the Rectangle: ");

        int height = sc.nextInt();

        float peri = 2 \* (base + height);

        return peri;

    }

}

public class CO3Q6 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        Circle c = new Circle();

        Rectangle r = new Rectangle();

        System.out.println("\n1. Area of the Circle");

        System.out.println("\n2. Perimeter of the Circle");

        System.out.println("\n3. Area of the Rectangle");

        System.out.println("\n4. Perimeter of the Rectangle");

        System.out.println("\n5. EXIT");

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

        int choice = sc.nextInt();

        switch (choice) {

            case 1:

                float ac = c.area();

                System.out.println("Area is " + ac);

                break;

            case 2:

                float pc = c.perimeter();

                System.out.println("Perimeter is " + pc);

                break;

            case 3:

                float ar = r.area();

                System.out.println("Area is " + ar);

                break;

            case 4:

                float pr = r.perimeter();

                System.out.println("Perimeter is " + pr);

                break;

            default:

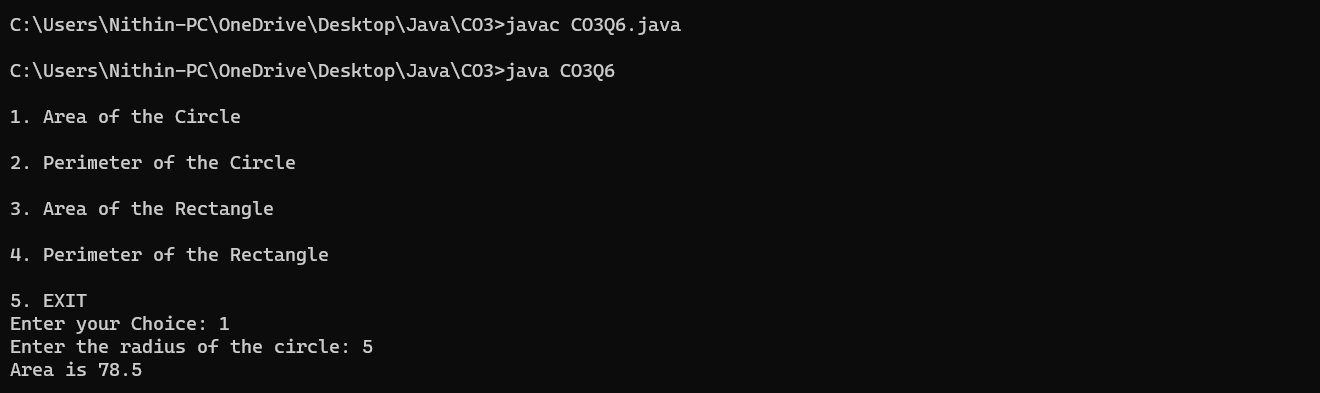
                System.out.println("Invalid choice");

        }

    }

}

**Output Screenshot**



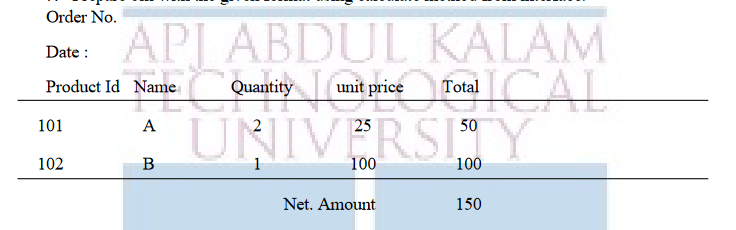
**Result**

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

**Experiment No.: 16**

**Aim**

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



**CO3**

Implement object-oriented concepts like inheritance, overloading and interfaces.

**Procedure**

import java.util.Scanner;

interface Bill {

    void inputs();

    void prints();

    double getTotal();

    double getNetAmount();

}

class SingleProduct implements Bill {

    Scanner sc = new Scanner(System.in);

    int productId, quantity, unitPrice;

    String name;

    double total;

    public void inputs() {

        System.out.print("Enter the Product Id: ");

        productId = sc.nextInt();

        System.out.print("Enter the Product Name: ");

        name = sc.next();

        System.out.print("Enter the Quantity: ");

        quantity = sc.nextInt();

        System.out.print("Enter the Unit Price: ");

        unitPrice = sc.nextInt();

        total = quantity \* unitPrice;

    }

    public void prints() {

        System.out.println(productId + "\t   " + name + "\t " + quantity + "\t     " + unitPrice + " \t " + total);

    }

    public double getTotal() {

        return total;

    }

    public double getNetAmount() {

        return getTotal();

    }

}

class BillGenerator implements Bill {

    int orderNo, num;

    String date;

    double amount;

    SingleProduct[] products;

    Scanner sc = new Scanner(System.in);

    public void inputs() {

        System.out.print("Enter the Order No: ");

        orderNo = sc.nextInt();

        System.out.print("Enter the date of purchase: ");

        date = sc.next();

        System.out.print("Enter the No of items: ");

        num = sc.nextInt();

        products = new SingleProduct[num];

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

            products[i] = new SingleProduct();

            products[i].inputs();

        }

    }

    public void prints() {

        System.out.println("Order No: " + orderNo);

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

        System.out.println("ProductID  Name  Quantity  UnitPrice     Total");

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

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

            products[i].prints();

        }

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

        System.out.println("\t\t\t\tNet Amount: " + getNetAmount());

    }

    public double getTotal() {

        double total = 0.0;

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

            total += products[i].getTotal();

        }

        return total;

    }

    public double getNetAmount() {

        return getTotal();

    }

}

public class CO3Q7 {

    public static void main(String[] args) {

        BillGenerator billGenerator = new BillGenerator();

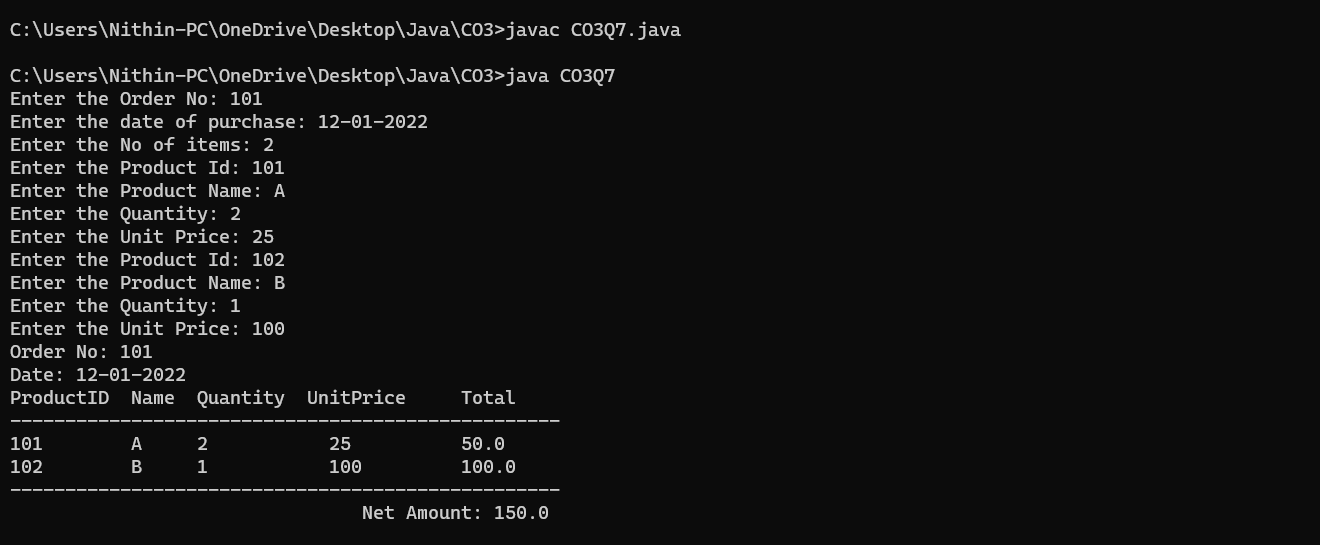
        billGenerator.inputs();

        billGenerator.prints();

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 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.

**CO4**

Implement packages, exception handling, multithreading and generic programming.

Use java.util package and Collection framework.

**Procedure**

package Graphics;

import java.util.\*;

interface Area {

    void Rectangle();

    void Triangle();

    void Square();

    void Circle();

}

class shapes implements Area {

    double lr, lb, ra, th, tb, ta, sa, cr, cc;

    public void getRect() {

        Scanner sc = new Scanner(System.in);

        System.out.print("\nEnter the length of the rectangle : ");

        lr = sc.nextDouble();

        System.out.print("\nEnter the breadth of the rectangle: ");

        lb = sc.nextDouble();

    }

    public void Rectangle() {

        ra = lr \* lb;

        System.out.print("\nArea of Rectangle is :" + ra);

    }

    public void getTri() {

        Scanner sc = new Scanner(System.in);

        System.out.print("\n\nEnter the height of the Triangle: ");

        th = sc.nextDouble();

        System.out.print("\nEnter the base of the Triangle: ");

        tb = sc.nextDouble();

    }

    public void Triangle() {

        ta = 0.5 \* th \* tb;

        System.out.print("\nArea of Triangle is " + ta);

    }

    public void getSqr() {

        Scanner sc = new Scanner(System.in);

        System.out.print("\n\nEnter the side of the Square: ");

        sa = sc.nextDouble();

    }

    public void Square() {

        sa = sa \* sa;

        System.out.print("\nArea of Square is: " + sa);

    }

    public void getCrl() {

        Scanner sc = new Scanner(System.in);

        System.out.print("\n\nEnter the radius of the Circle: ");

        cc = sc.nextDouble();

    }

    public void Circle() {

        cr = 3.14 \* cc \* cc;

        System.out.print("\nArea of Circle is: " + cr);

    }

}

public class CO4Q1 {

    public static void main(String[] args) {

        shapes o = new shapes();

        o.getRect();

        o.Rectangle();

        o.getTri();

        o.Triangle();

        o.getSqr();

        o.Square();

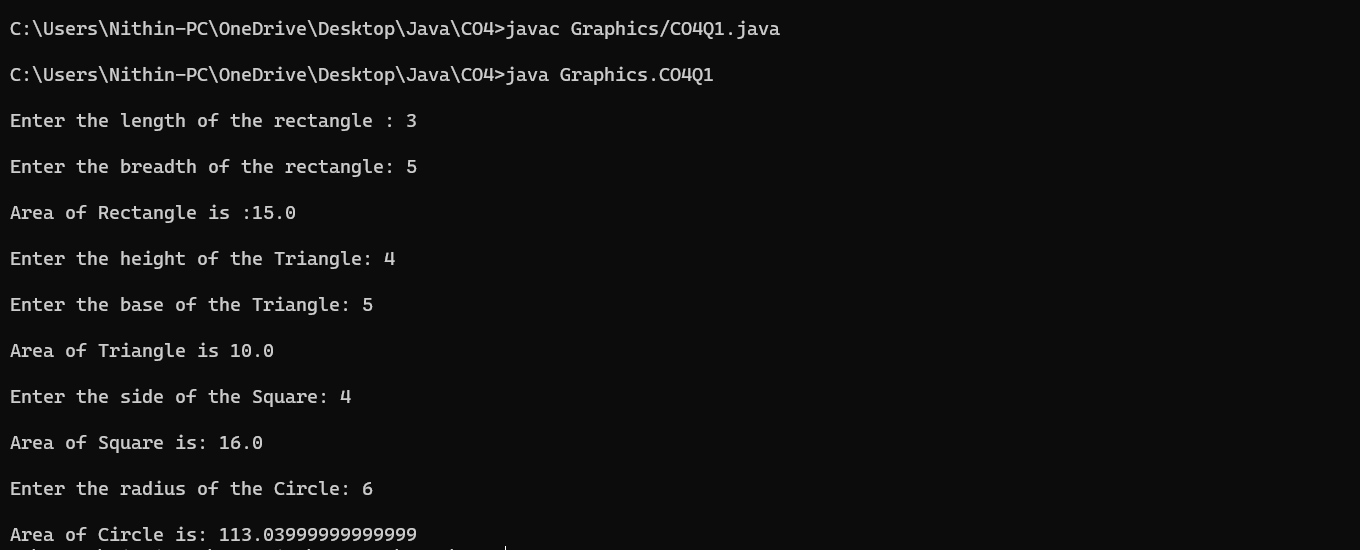
        o.getCrl();

        o.Circle();

    }

}

**Output Screenshot**



**Result**

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

**Experiment No.: 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.

**CO4**

Implement packages, exception handling, multithreading and generic programming.

Use java.util package and Collection framework.

**Procedure**

package Arithmetic;

interface Operation {

    double perform(double a, double b);

}

class Addition implements Operation {

    public double perform(double a, double b) {

        return a + b;

    }

}

class Subtraction implements Operation {

    public double perform(double a, double b) {

        return a - b;

    }

}

class Multiplication implements Operation {

    public double perform(double a, double b) {

        return a \* b;

    }

}

class Division implements Operation {

    public double perform(double a, double b) {

        if (b == 0) {

            System.out.println("Error: Division by zero is not allowed.");

            return 0;

        }

        return a / b;

    }

}

public class CO4Q2 {

    public static void main(String[] args) {

        double num1 = 10;

        double num2 = 5;

        Operation addition = new Addition();

        double sum = addition.perform(num1, num2);

        System.out.println("Sum: " + sum);

        Operation subtraction = new Subtraction();

        double difference = subtraction.perform(num1, num2);

        System.out.println("Difference: " + difference);

        Operation multiplication = new Multiplication();

        double product = multiplication.perform(num1, num2);

        System.out.println("Product: " + product);

        Operation division = new Division();

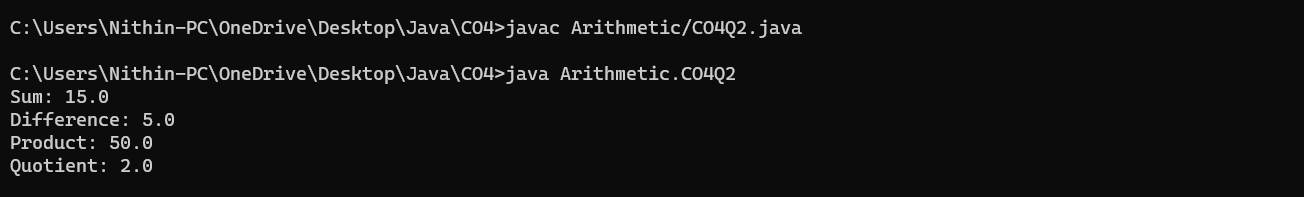
        double quotient = division.perform(num1, num2);

        System.out.println("Quotient: " + quotient);

    }

}

**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.

Use java.util package and Collection framework.

**Procedure**

import java.util.Scanner;

class NegativeNumberException extends Exception {

    public NegativeNumberException(String message) {

        super(message);

    }

}

public class CO4Q4 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the value of N: ");

        int n = scanner.nextInt();

        int sum = 0;

        int count = 0;

        System.out.println("Enter " + n + " positive integers:");

        for (int i = 0; i < n; i++) {

            try {

                int num = scanner.nextInt();

                if (num < 0) {

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

                }

                sum += num;

                count++;

            } catch (NegativeNumberException e) {

                System.out.println("Exception: " + e.getMessage());

                i--;

            }

        }

        if (count == 0) {

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

        } else {

            double average = (double) sum / count;

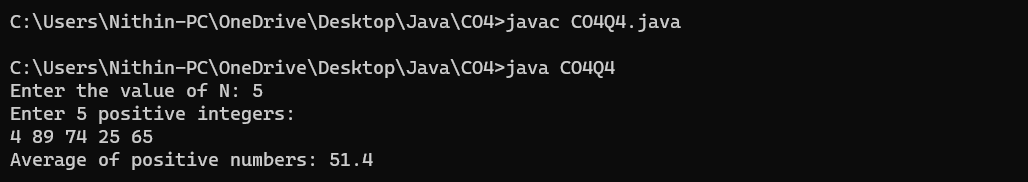
            System.out.println("Average of positive numbers: " + average);

        }

    }

}

**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.

Use java.util package and Collection framework

**Procedure**

import java.util.\*;

class MultiplicationTableThread extends Thread {

    public void run() {

        System.out.println("Multiplication Table of 5:");

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

            System.out.println("5 x " + i + " = " + (5 \* i));

        }

    }

}

class PrimeNumbersThread extends Thread {

    private int N;

    public PrimeNumbersThread(int N) {

        this.N = N;

    }

    public void run() {

        System.out.println("First " + N + " Prime Numbers:");

        int count = 0;

        int num = 2;

        while (count < N) {

            if (isPrime(num)) {

                System.out.print(num + " ");

                count++;

            }

            num++;

        }

    }

    private boolean isPrime(int number) {

        if (number <= 1) {

            return false;

        }

        for (int i = 2; i <= Math.sqrt(number); i++) {

            if (number % i == 0) {

                return false;

            }

        }

        return true;

    }

}

public class CO4Q5 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("\n Enter the value of N : ");

        int N = sc.nextInt();

        MultiplicationTableThread multiplicationTableThread = new MultiplicationTableThread();

        PrimeNumbersThread primeNumbersThread = new PrimeNumbersThread(N);

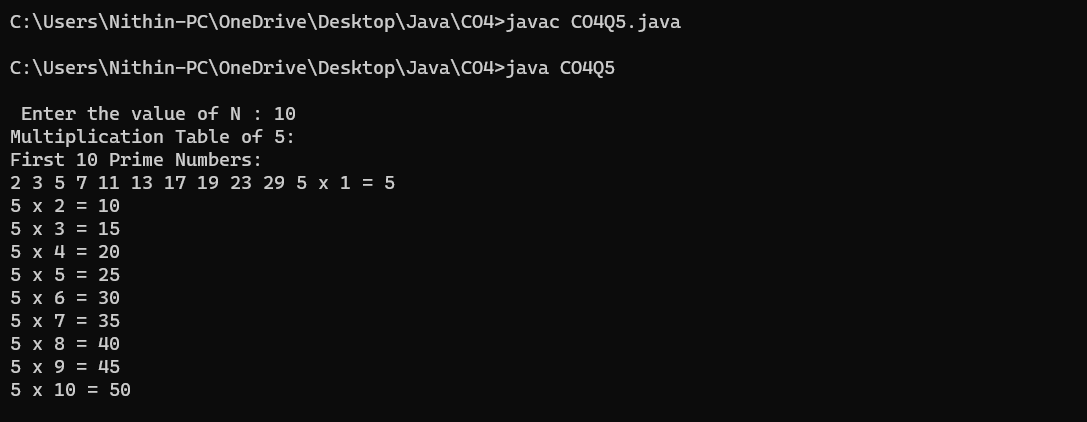
        multiplicationTableThread.start();

        primeNumbersThread.start();

    }

}

**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.

Use java.util package and Collection framework

**Procedure**

import java.util.Scanner;

class fibonacci implements Runnable {

    public void run() {

        int a1 = 0, a2 = 1, a3;

        Scanner sc = new Scanner(System.in);

        System.out.println("\n..GENERATING FIBINOCCI SERIES..");

        System.out.print("Enter the limit : ");

        int n = sc.nextInt();

        System.out.println("----FIBINOCCI SERIES----:");

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

            System.out.print(a1 + "->");

            a3 = a1 + a2;

            a1 = a2;

            a2 = a3;

        }

    }

}

class even implements Runnable {

    public void run() {

        Scanner sc = new Scanner(System.in);

        int upper, lower;

        System.out.println("\n..GENERATING EVEN SERIES..");

        System.out.print("Enter the lower limit :");

        lower = sc.nextInt();

        System.out.print("Enter the upper limit :");

        upper = sc.nextInt();

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

            if (i % 2 == 0) {

                System.out.print(i + "->");

            }

        }

    }

}

public class CO4Q6 {

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

        fibonacci fib = new fibonacci();

        Thread a = new Thread(fib);

        System.out.println("Thread Created. Enter the input in 5 sc");

        a.start();

        a.sleep(5000);

        even e = new even();

        Thread b = new Thread(e);

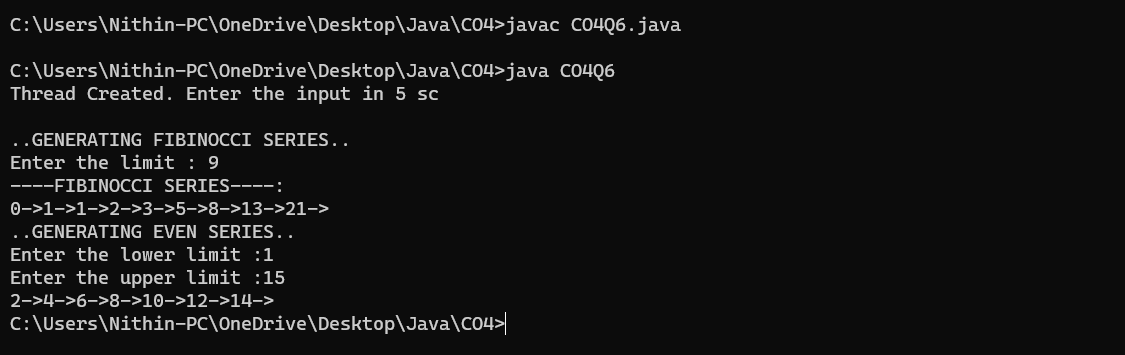
        b.run();

        b.sleep(5000);

    }

}

**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.

Use java.util package and Collection framework

**Procedure**

import java.util.Scanner;

public class CO4Q8 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        int ch;

        int top = -1;

        int n;

        System.out.print("Enter the size : ");

        n = sc.nextInt();

        int s[] = new int[n];

        do {

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

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

            System.out.println("3)PEEK ");

            System.out.println("4)DISPLAY ");

            System.out.println("5)IS EMPTY ");

            System.out.print("Select your Option : ");

            ch = sc.nextInt();

            switch (ch) {

                case 1:

                    if (top != n) {

                        System.out.print(" Enter the Element to be inserted : ");

                        s[++top] = sc.nextInt();

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

                    } else {

                        System.out.println("Stack is Full");

                    }

                    break;

                case 2:

                    if (top == -1) {

                        System.out.println("Stack Under Flow");

                    } else {

                        top = top - 1;

                        System.out.println("One Element is deleted");

                    }

                    break;

                case 3:

                    System.out.println("TOP Of Stack is :" + s[top]);

                    break;

                case 5:

                    if (top == -1) {

                        System.out.println("Stack Is Empty!!");

                    } else {

                        System.out.println("Stack is not Empty !!");

                    }

                    break;

                case 4:

                    if (top == -1) {

                        System.out.println("Stack Is Empty!!");

                    } else {

                        System.out.println("Stack Elements are!!");

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

                            System.out.print(s[i] + "\_\_\_");

                        }

                    }

                    break;

            }

            System.out.println("\n1.Exit ");

            System.out.println("0.Continue");

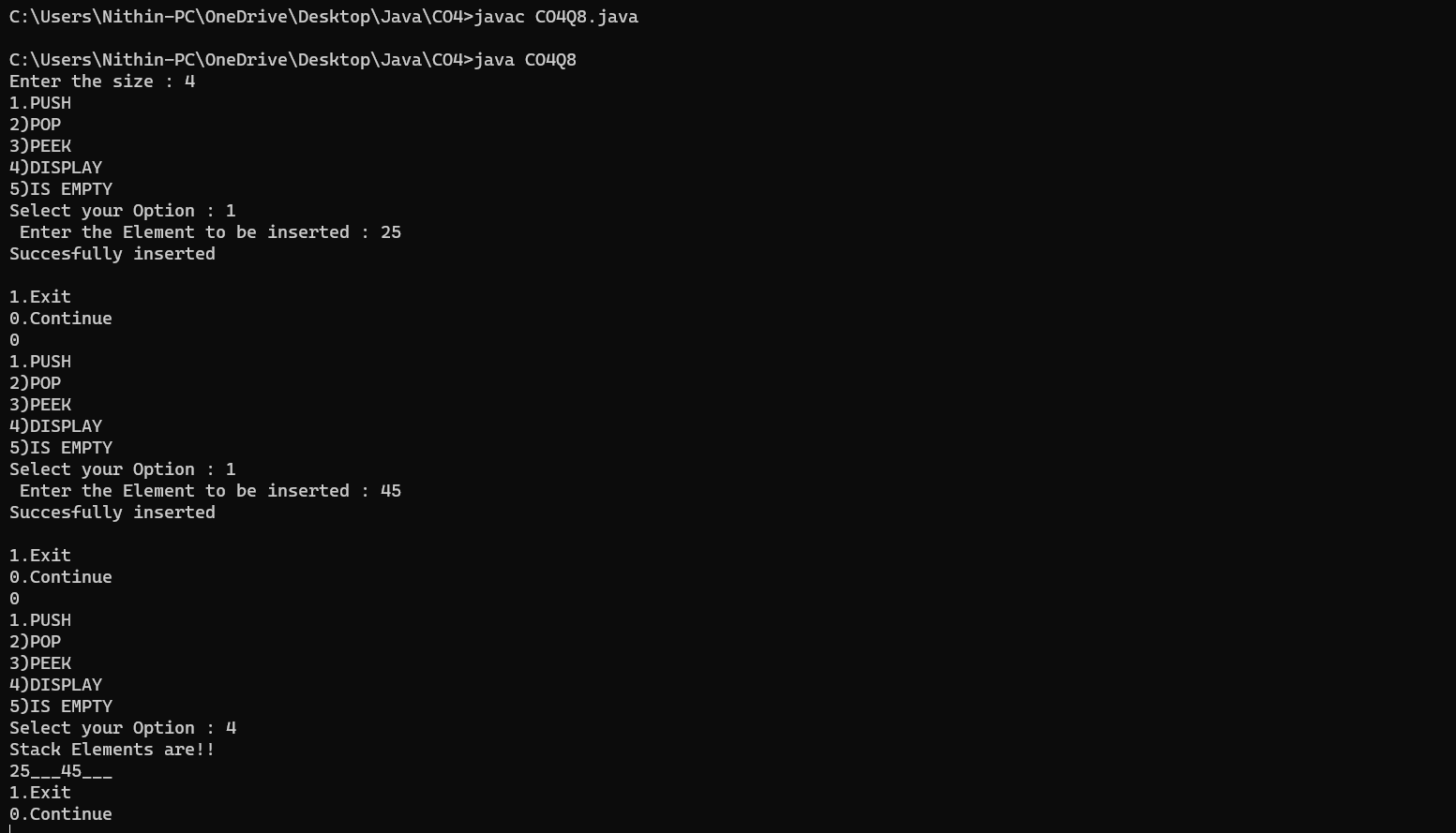
            ch = sc.nextInt();

        } 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.

Use java.util package and Collection framework

**Procedure**

import java.util.Scanner;

class Bubble {

    public void sort(int a[], int n) {

        int t;

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

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

                if (a[j] > a[j + 1]) {

                    t = a[j];

                    a[j] = a[j + 1];

                    a[j + 1] = t;

                }

            }

        }

    }

}

public class CO4Q9 {

    public static void main(String args[]) {

        int n;

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the size : ");

        n = sc.nextInt();

        int b[] = new int[n];

        for (int i = 0; i < n; i++) {

            System.out.print("Enter the element" + (i + 1) + " :");

            b[i] = sc.nextInt();

        }

        System.out.println("\nArray before sort");

        for (int i = 0; i < n; i++) {

            System.out.print(b[i] + "\_\_");

        }

        Bubble bs = new Bubble();

        bs.sort(b, n);

        System.out.print("\n\nArray after sort\n");

        for (int i = 0; i < n; i++) {

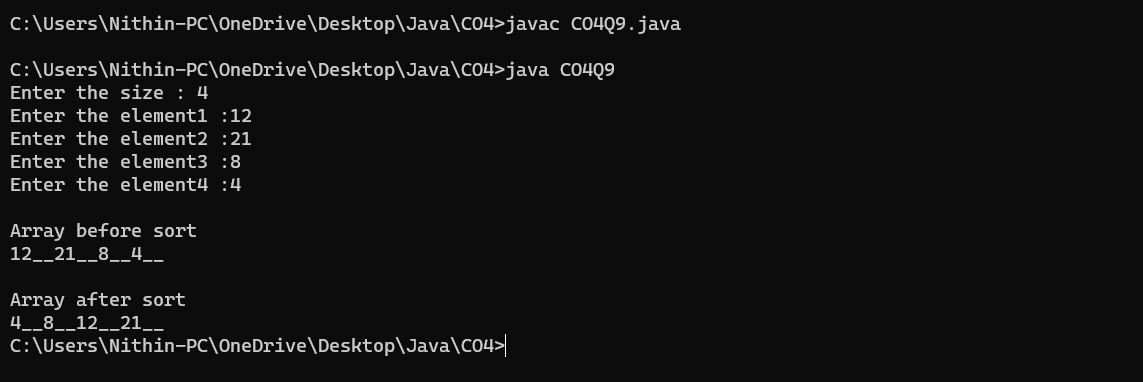
            System.out.print(b[i] + "\_\_");

        }

    }

}

**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.

Use java.util package and Collection framework

**Procedure**

import java.util.Scanner;

import java.util.ArrayList;

public class CO4Q10 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        int ch, n;

        String str;

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

        do {

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

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

            System.out.println("3. Display the element in n\_th index ");

            System.out.println("4. DISPLAY ");

            System.out.println("5. Remove all ");

            System.out.print("Select your Option  :");

            ch = sc.nextInt();

            switch (ch) {

                case 1:

                    System.out.print("Enter the element to be inserted : ");

                    str = sc.next();

                    list.add(str);

                    break;

                case 2:

                    System.out.print("Enter index of the element  to be removed : ");

                    n = sc.nextInt();

                    list.remove(n);

                    break;

                case 3:

                    System.out.print("Enter the index of element : ");

                    n = sc.nextInt();

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

                    break;

                case 4:

                    System.out.println(list);

                    break;

                case 5:

                    list.removeAll(list);

                    System.out.print("Succesfully Removed all elements ");

                    break;

            }

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

            System.out.println("0.Continue");

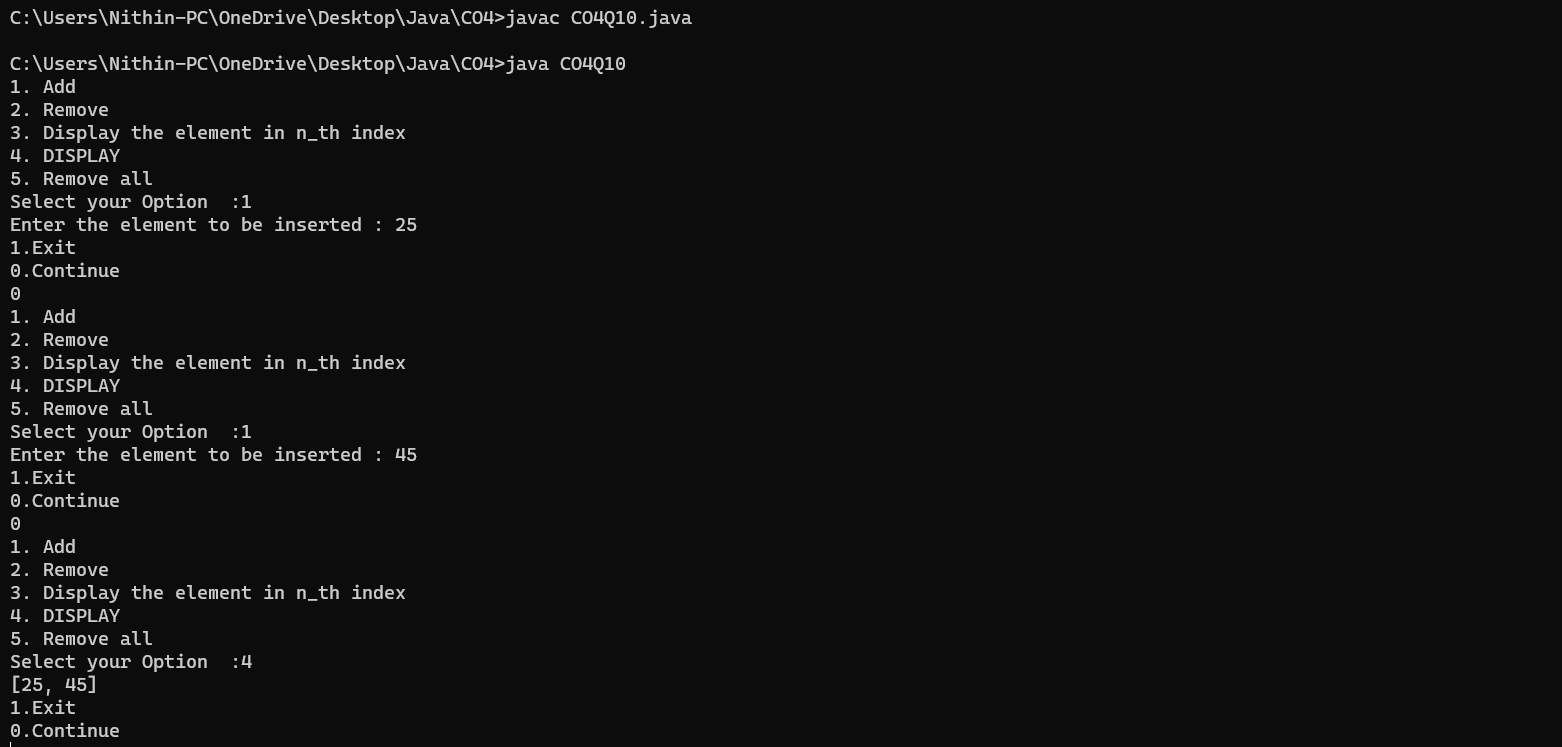
            ch = sc.nextInt();

        } while (ch == 0);

    }

}

**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.

Use java.util package and Collection framework.

**Procedure**

import java.util.\*;

public class CO4Q11 {

    public static void main(String[] args) {

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

        Scanner sc = new Scanner(System.in);

        int ch;

        do {

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

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

            System.out.println("3. Clear ");

            System.out.println("4. Display ");

            System.out.print("Select your Option : ");

            ch = sc.nextInt();

            switch (ch) {

                case 1:

                    String val;

                    int v;

                    System.out.print("Enter the element : ");

                    val = sc.next();

                    list.add(val);

                    break;

                case 2:

                    System.out.println("Element to be removed :");

                    v = sc.nextInt();

                    list.remove(v);

                    break;

                case 3:

                    list.clear();

                    System.out.println("All element are removed ");

                    break;

                case 4:

                    System.out.println(list);

                    break;

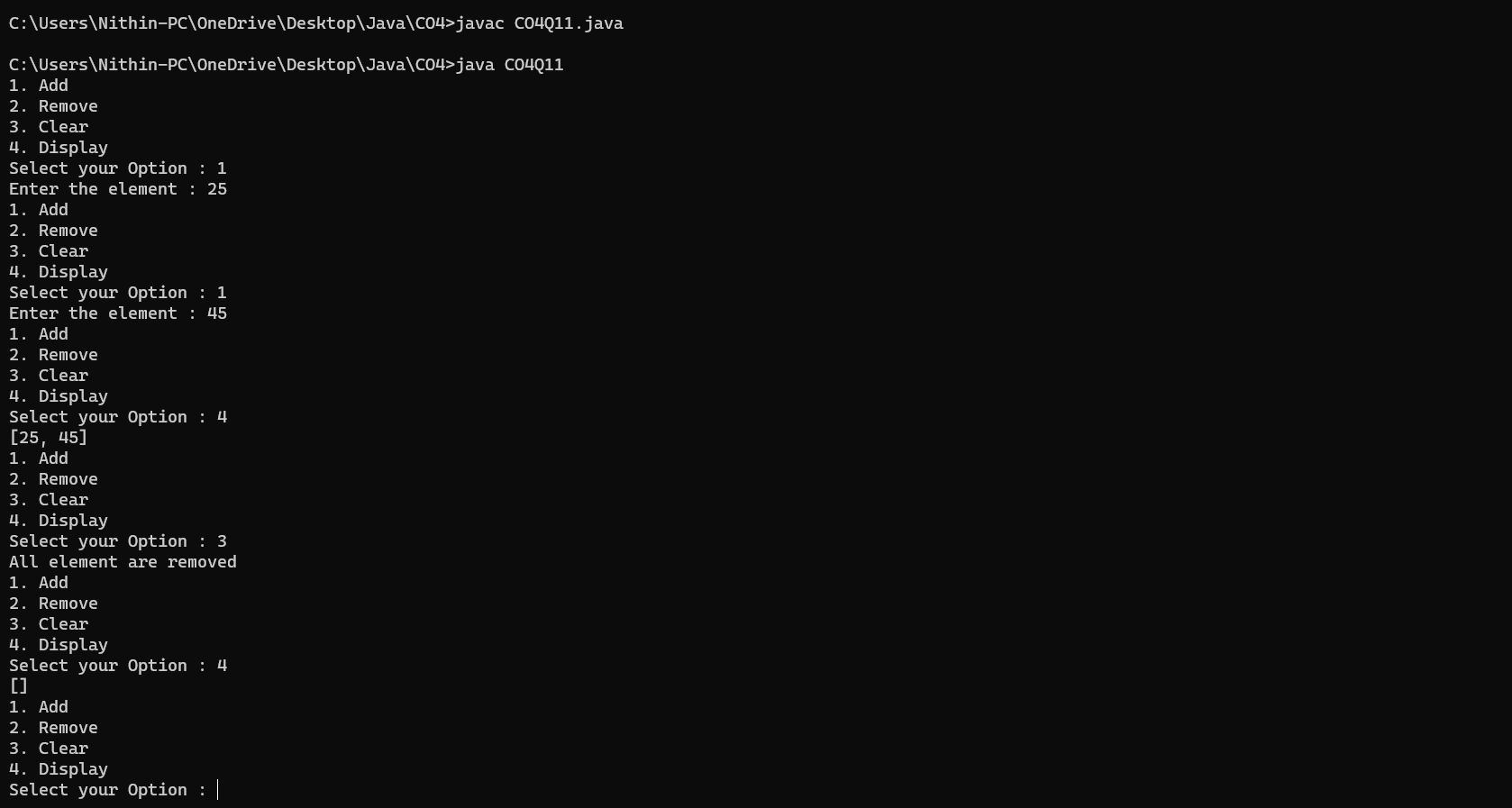
            }

        } while (ch != 0);

    }

}

**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.

Use java.util package and Collection framework.

**Procedure**

import java.util.Scanner;

import java.util.Stack;

public class CO4Q12 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the length of the stack: ");

        int length = sc.nextInt();

        sc.nextLine();

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

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

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

            String element = sc.nextLine();

            stack.push(element);

        }

        System.out.print("Enter the index position to remove (1 - " + stack.size() + "): ");

        int positionToRemove = sc.nextInt();

        removeElementAtPosition(stack, positionToRemove);

        System.out.println("Stack after removing element at position " + positionToRemove + ":");

        System.out.println(stack);

    }

    public static void removeElementAtPosition(Stack<String> stack, int position) {

        if (position < 1 || position > stack.size()) {

            System.out.println("Invalid position. No element removed.");

            return;

        }

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

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

            tempStack.push(stack.pop());

        }

        stack.pop();

        while (!tempStack.isEmpty()) {

            stack.push(tempStack.pop());

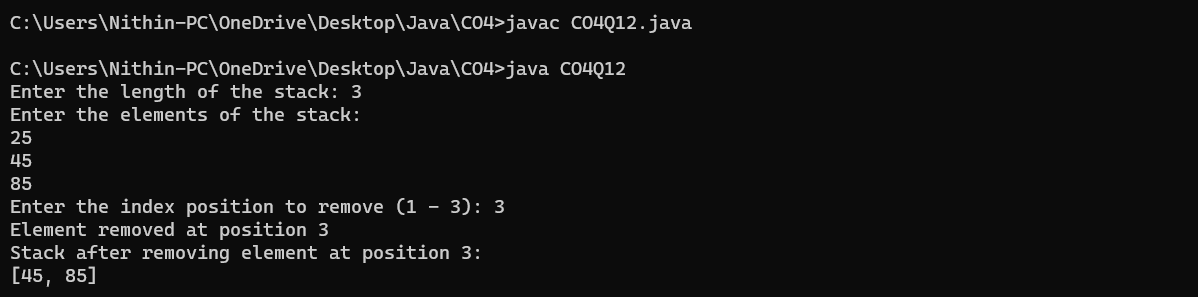
        }

        System.out.println("Element removed at position " + position);

    }

}

**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.

Use java.util package and Collection framework.

**Procedure**

import java.util.\*;

class CO4Q13 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

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

        int ch;

        do {

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

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

            System.out.println("3. Poll");

            System.out.print("Select your option: ");

            ch = sc.nextInt();

            switch (ch) {

                case 1:

                    int val;

                    System.out.print("Enter the element : ");

                    val = sc.nextInt();

                    pq.add(val);

                    break;

                case 2:

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

                    break;

                case 3:

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

                    break;

            }

        } while (ch != 0);

    }

}

**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.

Use java.util package and Collection framework.

**Procedure**

import java.util.\*;

import java.util.Deque;

public class CO4Q14 {

    public static void main(String[] args) {

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

        dq.addFirst(1);

        dq.addLast(2);

        int frist = dq.removeFirst();

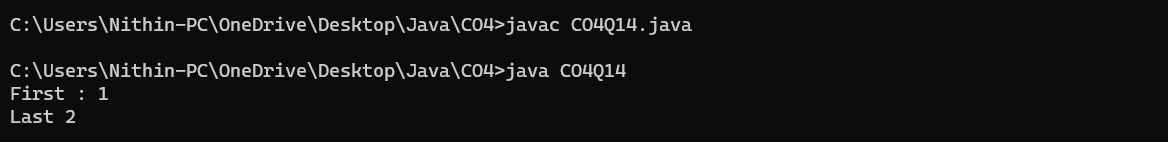
        int last = dq.removeLast();

        System.out.println("First : " + frist + "\nLast " + 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.

Use java.util package and Collection framework.

**Procedure**

import java.util.\*;

public class CO4Q15 {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        Set<Integer> Srt = new LinkedHashSet<>();

        Set<Integer> Srt2 = new LinkedHashSet<>();

        Set<Integer> Union = new LinkedHashSet<>();

        Set<Integer> Inter = new LinkedHashSet<>();

        int n, u, l;

        System.out.print("Enter the Number of elements Set1 : ");

        n = sc.nextInt();

        for (int i = 0; i < n; i++) {

            System.out.print("Enter the element " + (i + 1) + " : ");

            u = sc.nextInt();

            Srt.add(u);

        }

        System.out.println(Srt);

        System.out.print("Enter the Number of elements Aet2 : ");

        l = sc.nextInt();

        for (int i = 0; i < l; i++) {

            System.out.print("Enter the element " + (i + 1) + " : ");

            u = sc.nextInt();

            Srt2.add(u);

        }

        System.out.println(Srt2);

        // Union

        Union.addAll(Srt);

        boolean unionChanged = Union.addAll(Srt2);

        System.out.println("\nUnion : " + Union);

        System.out.println("New elements were added to set1 on union: " + unionChanged);

        // Intersection

        Inter.addAll(Srt);

        boolean intersectionChanged = Inter.retainAll(Srt2);

        System.out.println("\nIntersection : " + Inter);

        System.out.println("Elements were removed from the set1 on intersection: " + intersectionChanged);

        // Difference

        Set<Integer> difference = new LinkedHashSet<>(Srt);

        boolean differenceChanged = difference.removeAll(Srt2);

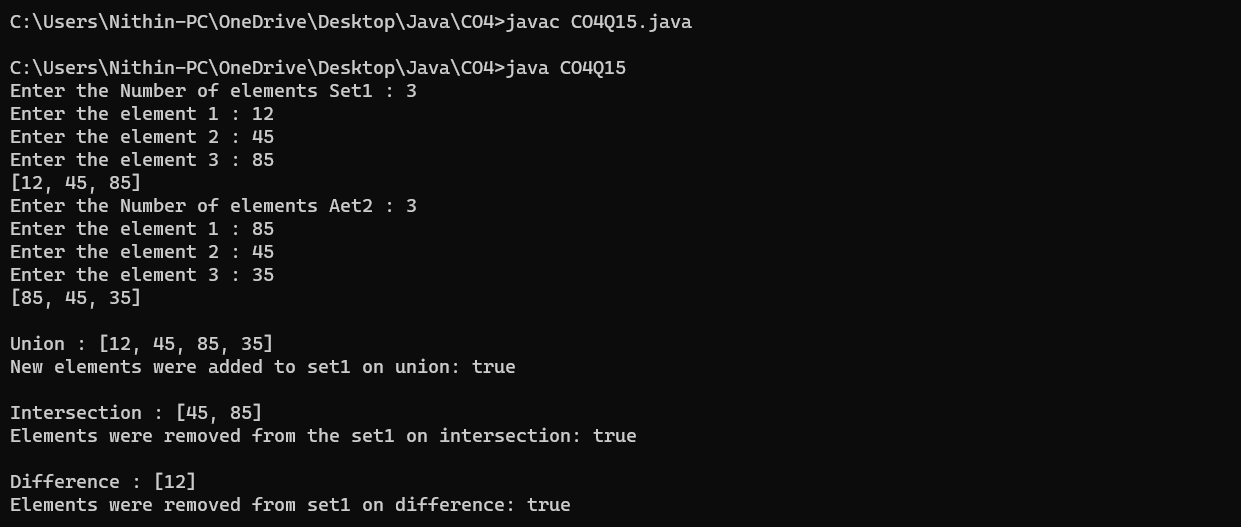
        System.out.println("\nDifference : " + difference);

        System.out.println("Elements were removed from set1 on difference: " + differenceChanged);

    }

}

**Output Screenshot**



**Result**

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