20MCA136 - NETWORKING &

SYSTEM ADMINISTRATION LAB

*Lab Report Submitted By*

**SANDRA P M**

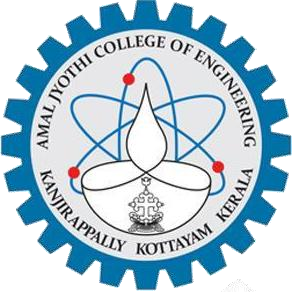
**Reg. No.: AJC21MCA-2092**

*In Partial fulfillment for the Award of the Degree Of*

**MASTER OF COMPUTER APPLICATIONS (2 Year)**

**(MCA)**

**APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY**



**AMAL JYOTHI COLLEGE OF ENGINEERING KANJIRAPPALLY**

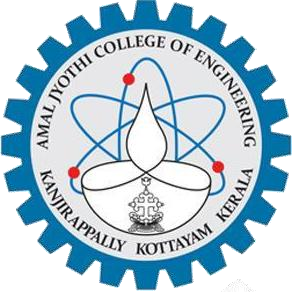
## [Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with ‘A’ grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

**2021-2023**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**AMAL JYOTHI COLLEGE OF ENGINEERING**

**KANJIRAPPALLY**



**CERTIFICATE**

This is to certify that the Lab report, **“20MCA132 OBJECT ORIENTED PROGRAMMING LAB”** is the bonafide work of **Sandra P M AJC21MCA-2092** in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications under APJ Abdul Kalam Technological University during the year 2021-22.

**Rev.Fr.Dr.Rubin Thottupuram Jose**

**Lab In-Charge**

**Gloriya Mathew Head of the Department**

**Internal Examiner External Examiner**

**CONTENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl.No** | **Content** | **Date** | **Page No** |
| **1** |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Name: Sandra PM**

**Roll No:34**

**Batch:MCA**

**Date:5/04/22**

**Object oriented programming lab**

**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.

**Source code:**

class Product{

String pcode,pname;

double price;

void details()

{

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

System.out.println("PCode:"+pcode);

System.out.println("PName:"+pname);

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

}

}

public class ProductDetails{

public static void main (String args [])

{

Product p1 = new Product();

p1.pcode = "13wtno1";

p1.pname = "camlin notebook";

p1.price = 45;

System.out.println("\nProduct1:");

p1.details();

Product p2 = new Product();

p2.pcode = "13wtno2";

p2.pname = "pinpoint pen";

p2.price = 10;

System.out.println("\nProduct2:");

p2.details();

Product p3 = new Product();

p3.pcode = "13wtno3";

p3.pname = "classmates notebook";

p3.price = 50;

System.out.println("\nProduct3:");

p3.details();

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

{

System.out.println("\n\nProduct with lowest price is:");

p1.details();

}

else if(p2.price<p3.price)

{

System.out.println("\n\nProduct with lowest price is:");

p2.details();

}

else

{

System.out.println("\n\nProduct with lowest price is:");

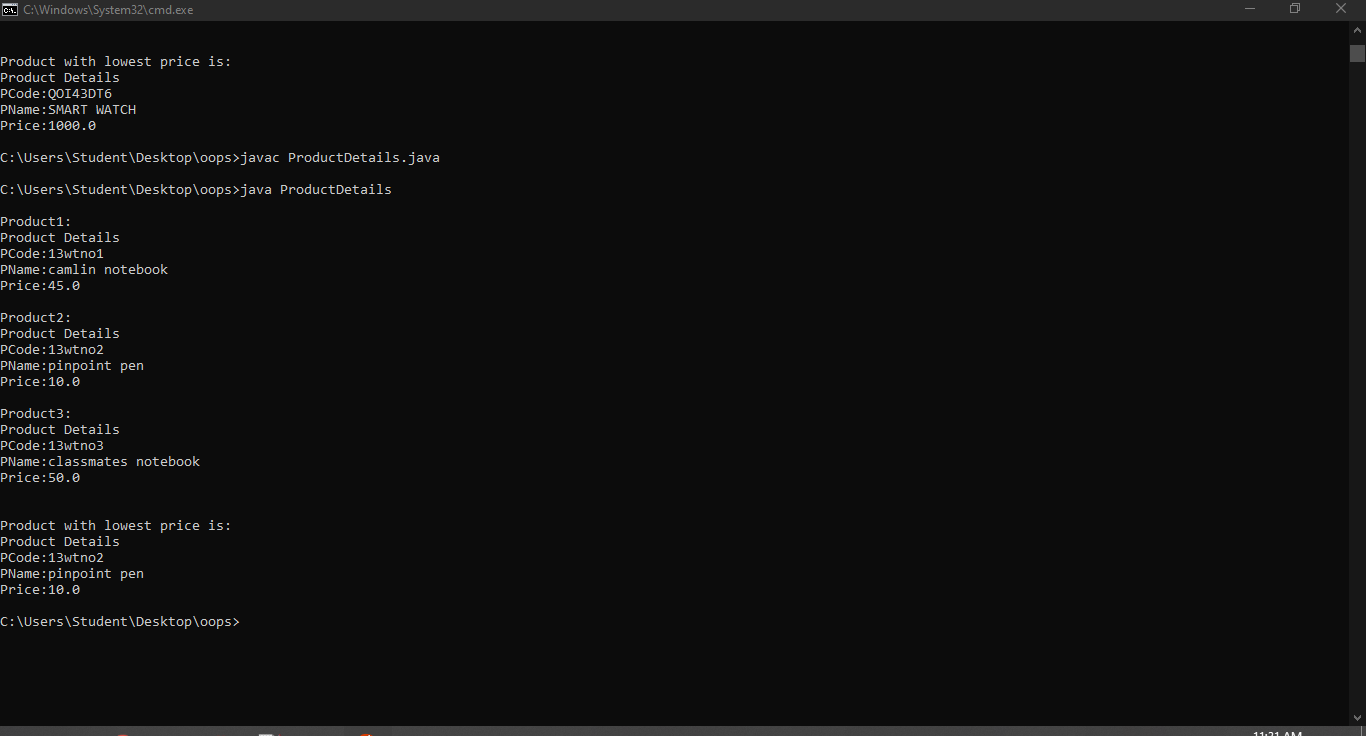
p3.details();

}

}

}

**output:**

****

**Name: Sandra PM**

**Roll No:34**

**Batch:MCA**

**Date:5/04/22**

**Object oriented programming lab**

**Experiment No.: 2**

**Aim**

Read 2 matrices from the console and perform matrix addition.

**Source code:**

import java.util.Scanner;

class AddMatrix {

public static void main (String args[]){

int row,col,i,j;

Scanner in =new Scanner(System.in);

System.out.println("enter the rows");

row =in.nextInt();

System.out.println("enter the columns");

col=in.nextInt();

int mat1[][]=new int [row][col];

int mat2[][]=new int [row][col];

int res[][]=new int[row][col];

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

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

{

for(j=0;j<col;j++)

mat1[i][j]=in.nextInt();

System.out.println();

}

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

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

{

for(j=0;j<col;j++)

{

mat2[i][j]=in.nextInt();

System.out.println();

}

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

for(j=0;j<col;j++)

res[i][j]=mat1[i][j]+mat2[i][j];

System.out.println("sum of matrices");

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

{

for(j=0;j<col;j++)

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

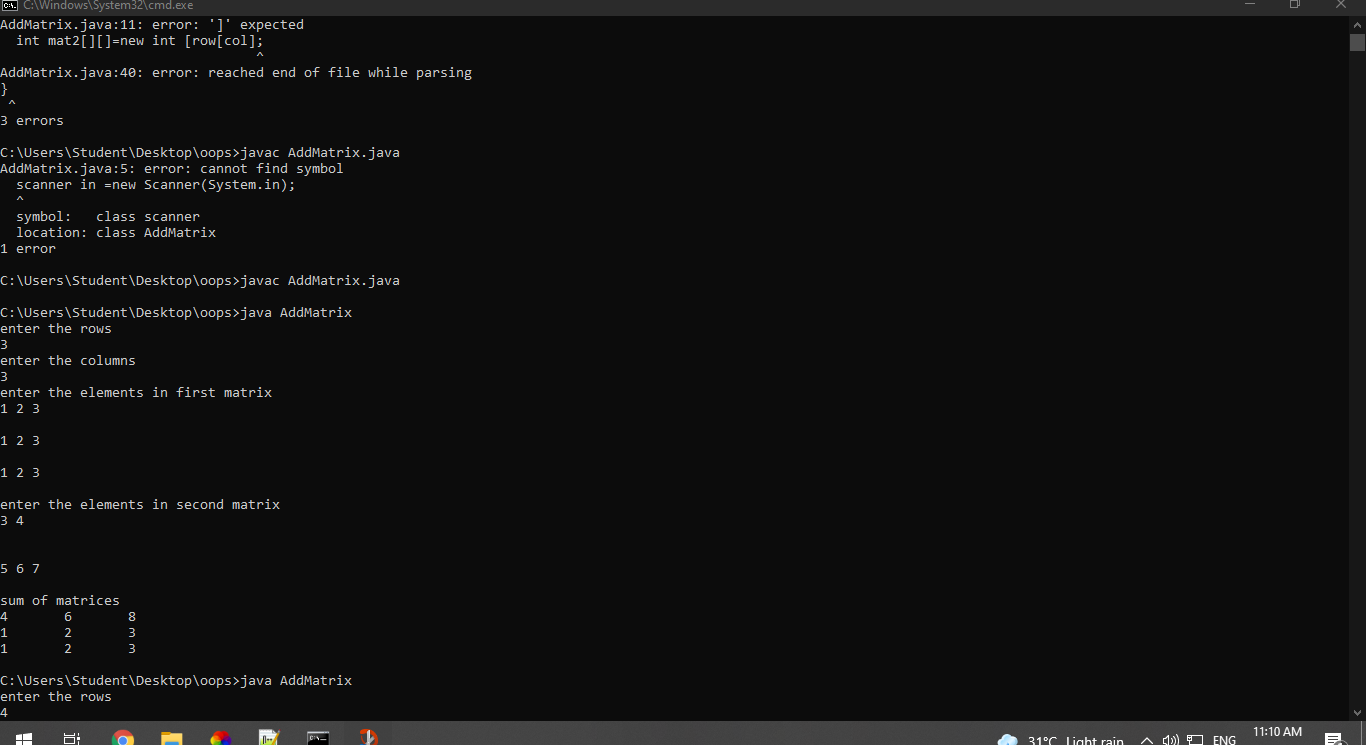
System.out.println();

}

}

}

}

**Output: **

**Name: Sandra P M**

**Roll No:34**

**Batch:MCA B**

**Date:06-04-2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 3**

**Aim**

Add complex numbers

**Procedure**

import java.util.\*;

class Complex {

int real, imaginary;

Complex(){}

Complex(int tempReal, int tempImaginary){

real = tempReal;

imaginary = tempImaginary;

}

Complex addComp(Complex C1, Complex C2){

Complex temp = new Complex();

temp.real = C1.real + C2.real;

temp.imaginary = C1.imaginary + C2.imaginary;

return temp;

}

Complex subtractComp(Complex C1, Complex C2){

Complex temp = new Complex();

temp.real = C1.real - C2.real;

temp.imaginary = C1.imaginary - C2.imaginary;

return temp;

}

void printComplexNumber(){

System.out.println("Complex number: "

+ real + " + "

+ imaginary + "i");

}

}

public class ComplexNumber {

public static void main(String[] args){

Complex C1 = new Complex(5, 2);

C1.printComplexNumber();

Complex C2 = new Complex(8, 7);

C2.printComplexNumber();

Complex C3 = new Complex();

C3 = C3.addComp(C1, C2);

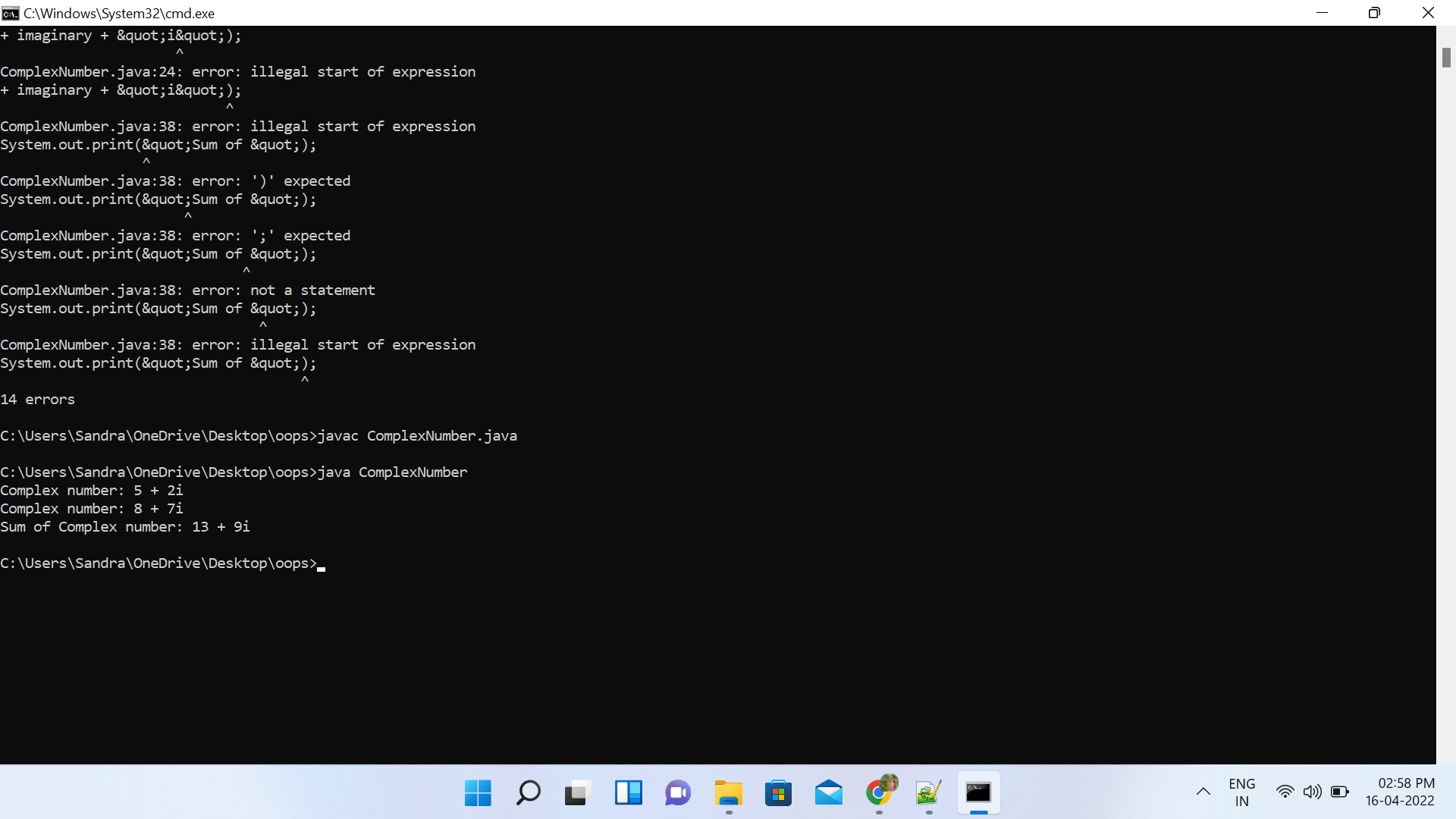
System.out.print("Sum of ");

C3.printComplexNumber();

}

**}**

**Output**

****

**Name: Sandra P M**

**Roll No:34**

**Batch: MCA B**

**Date:06/04/22**

**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 4**

**Aim**

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

**Procedure**

import java.util.\*;

public class Symetric {

static void checkSymmetric(int mat[][], int row,int col){

int i, j, flag = 1;

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

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

for (j = 0; j < col; j++) {

System.out.print(mat[i][j] + "\t");

}

System.out.println("");

}

int[][] transpose = new int[row][col];

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

for (j = 0; j < col; j++) {

transpose[j][i] = mat[i][j];

}

}

if (row == col) {

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

for (j = 0; j <col; j++) {

if (mat[i][j] != transpose[i][j]) {

flag = 0;

break;

}

}

if (flag == 0) {

System.out.print("\nThe matrix is not symmetric");

break;

}

}

if (flag == 1) {

System.out.print("\nThe matrix is symmetric");

}

}

else {

System.out.print("\nThe matrix is not symmetric");

}

}

public static void main(String args[]){

Scanner sc = new Scanner(System.in);

int i, j, row, col, flag = 1;

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

row = sc.nextInt();

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

col = sc.nextInt();

int[][] mat = new int[row][col];

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

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

for (j = 0; j<col; j++) {

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

}

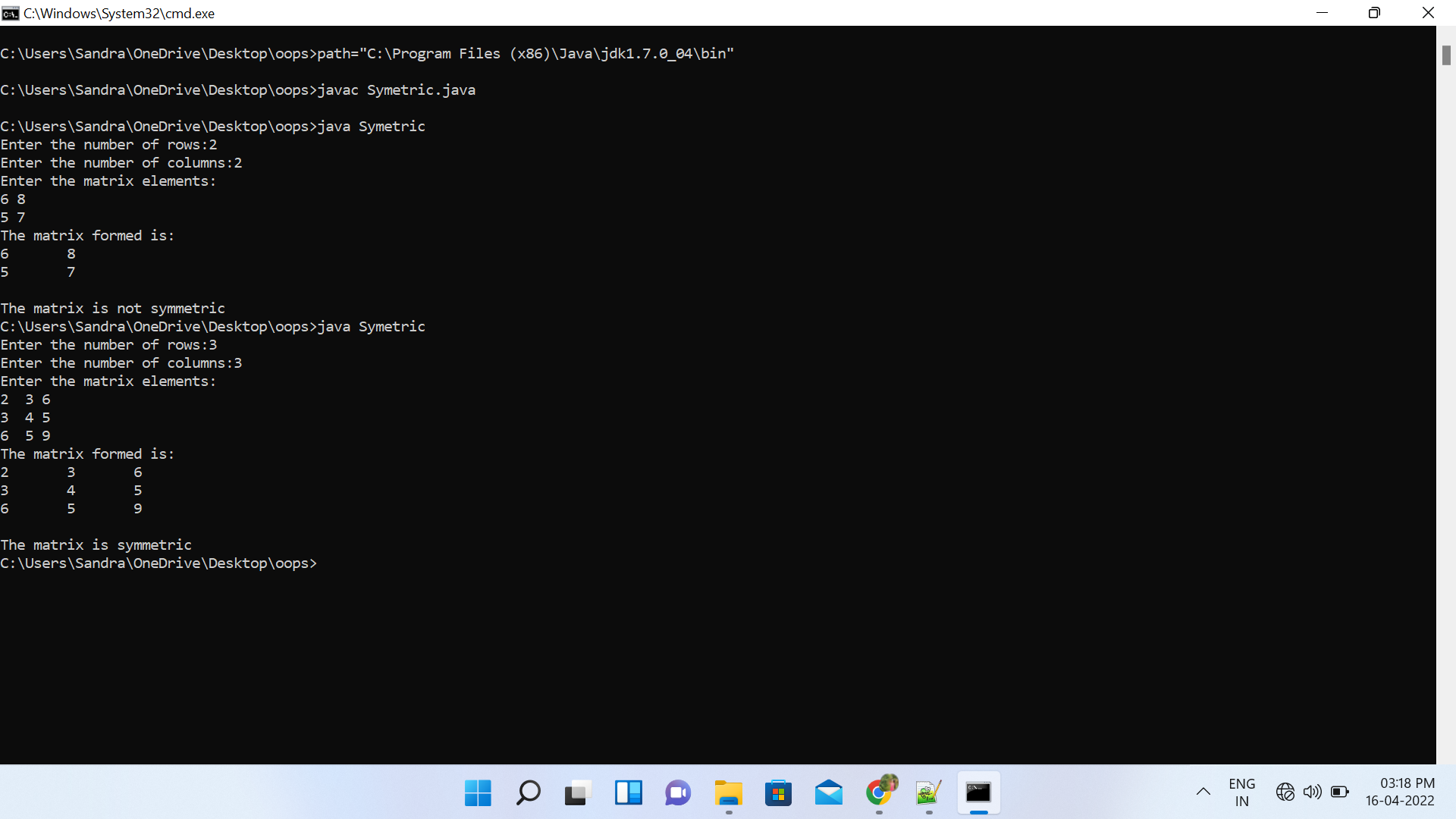
}

checkSymmetric(mat, row, col);

}

}

**Output**



**Name: Sandra P M**

**Roll No:34**

**Batch: MCA B**

**Date:17-05-2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**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.

**Procedure**

class Cpu

{

int price=5000;

class Processor

{

int nop=10;

String manf="Simon Woods";

}

static class Ram

{

static String memory="6GB";

String manf="intel";

}

}

public class Computer

{

public static void main (String args[])

{

Cpu obj1=new Cpu();

Cpu.Processor obj2=obj1.new Processor();

Cpu.Ram obj3= new Cpu.Ram();

System.out.println("Cpu price:"+obj1.price);

System.out.println("Number of processors:"+obj2.nop);

System.out.println("Processor Manufacturer"+obj2.manf);

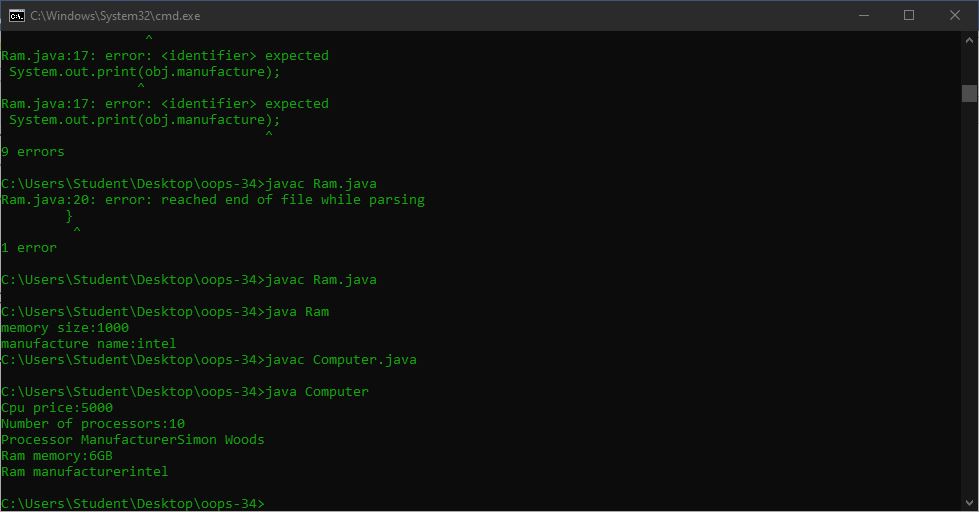
System.out .println("Ram memory:"+ Cpu.Ram.memory);

System.out.println("Ram manufacturer"+obj3.manf);

}

}

Output



**Name: Sandra PM**

**Roll No:34**

**Batch: MCA b**

**Date:22/04/2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 6**

**Aim**

**Program to Sort strings**

**Procedure**

import java.util.Arrays;

public class Sort{

public static void main(String args[])

{

String [] names={"san","saju","sarangi"};

System.out.println("array names:"+Arrays.toString(names));

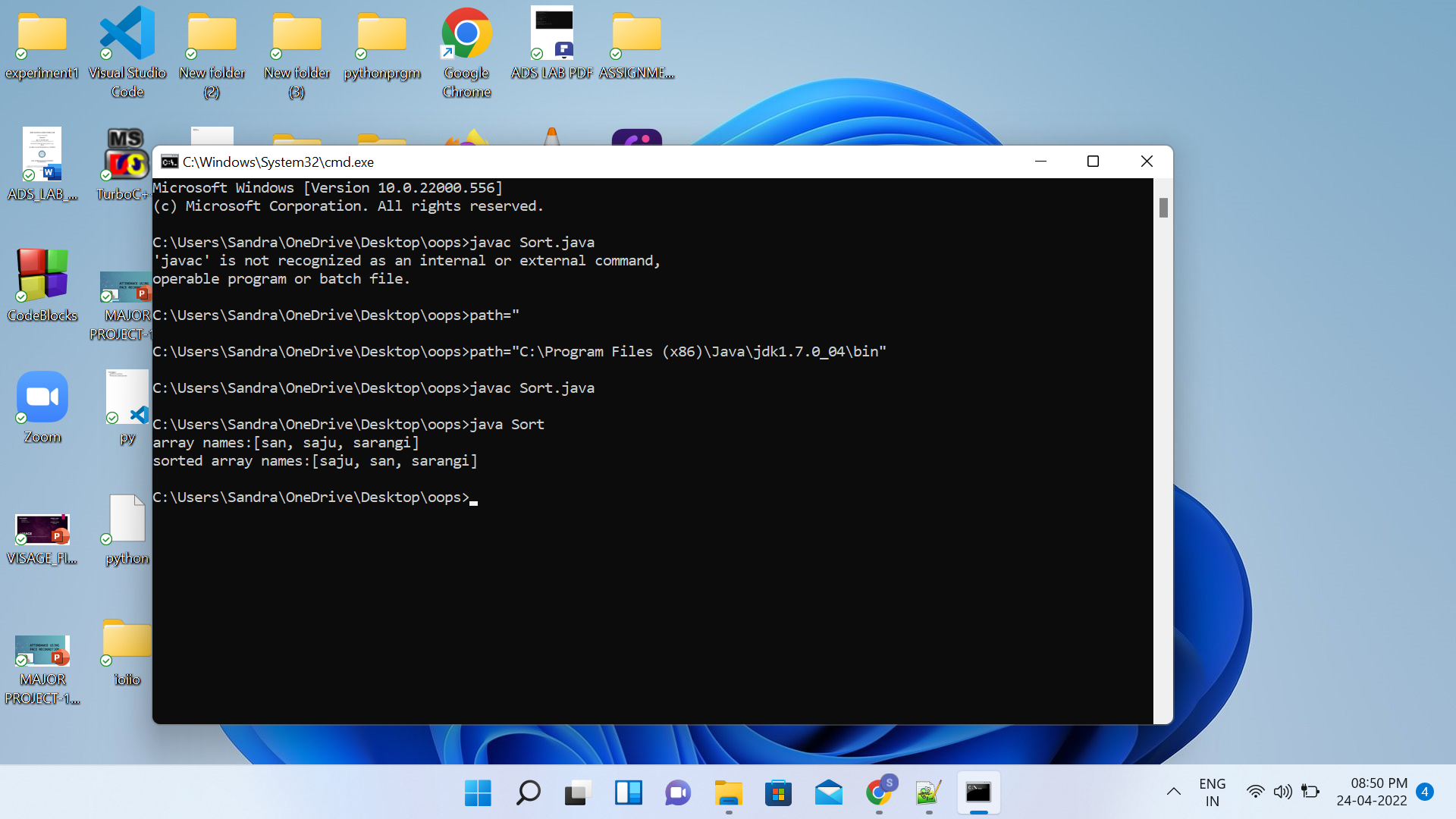
Arrays.sort(names);

System.out.println("sorted array names:"+Arrays.toString(names));

}

}

**Output Screenshot**



**Name: Sandra PM**

**Roll No:34**

**Batch: MCA B**

**Date:22/04/2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 7**

**Aim**

Search an element in an array.

**Procedure**

import java.util.Scanner;

public class Searcharray

{

public static void main(String[] args)

{

int n, element, flag = 0, i = 0;

Scanner s = new Scanner(System.in);

System.out.print("Enter no. of elements you want in array:");

n = s.nextInt();

int a[] = new int[n];

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

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

{

a[i] = s.nextInt();

}

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

element = s.nextInt();

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

{

if(a[i] == element)

{

flag = 1;

break;

}

else

{

flag = 0;

}

}

if(flag == 1)

{

System.out.println("Element found at position:"+(i + 1)+""+"\n"+"searched element is:"+element);

}

else

{

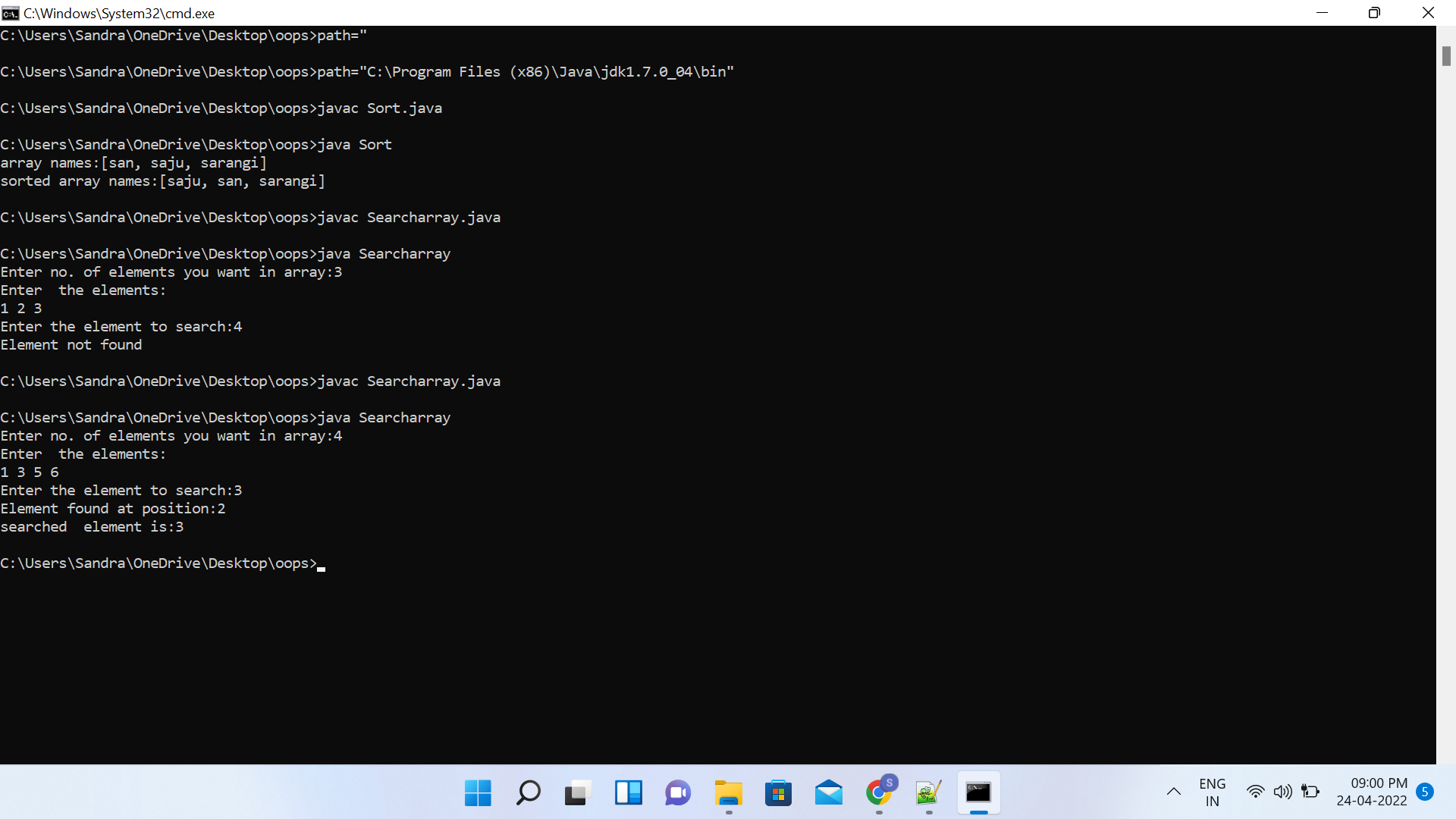
System.out.println("Element not found");

}

}

}

**Output Screenshot**



**Name:Sandra P M**

**Roll No:34**

**Batch:MCA B**

**Date:22/04/22**

**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 8**

**Aim**

Perform string manipulations

**Procedure**

public class StringManipulation {

public static void main(String[] args){

String str1= "This is my first job ", str2="and I like it.";

System.out.println("The string 01 is : "+str1+"\nString 02 is : "+str2);

String strconcat= str1+str2;

System.out.println("\nThe concatenation of two strings is : "+strconcat);

String strUppercase= str1.toUpperCase();

System.out.println("\nNormal String to uppercase string is : "+strUppercase);

String strLowercase= str2.toLowerCase();

System.out.println("\nNormal String to lowercase string is : "+strLowercase);

String strsubString= str1.substring(5);

System.out.println("\nSubstring of the string is : "+strsubString);

String strtrim= str1.trim();

System.out.println("\nString trim is given by : "+strtrim);

boolean strcontains= str1.contains("my");

System.out.println("\nCheck if the string 01 contains -'my' : "+strcontains);

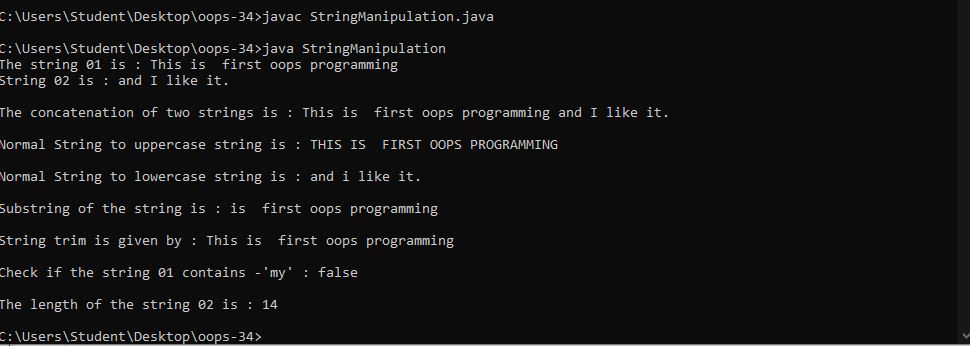
int strlength= str2.length();

System.out.println("\nThe length of the string 02 is : "+strlength);

}

}

**Output**



**Object oriented programming lab**

**Name: sandra pm**

**Roll No:34**

**Batch: mca-b**

**Date:17-5-2022**

**Experiment No.: 10**

**Aim**

Area of different shapes using overloaded functions

**Procedure**

**c**lass OverloadDemo

{

void area(float x)

{

System.out.println("the area of the square is "+Math.pow(x, 2)+" sq units");

}

void area(float x, float y)

{

System.out.println("the area of the rectangle is "+x\*y+" sq units");

}

void area(double x)

{

double z = 3.14 \* x \* x;

System.out.println("the area of the circle is "+z+" sq units");

}

}

class Overload

{

public static void main(String args[])

{

OverloadDemo obj = new OverloadDemo();

ob.area(8);

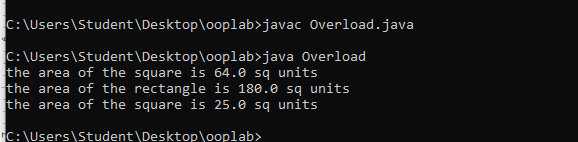
ob.area(12,15);

ob.area(5);

}

}

**Output Screenshot**

****

**Object oriented programming lab**

**Name: sandra pm**

**Roll No:34**

**Batch: mca-b**

**Date:17-5-2022**

**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.

**Procedure**

class EMPS{

public static void main(String[] args) {

Teacher tobj[] = new Teacher[2];

tobj[0] = new Teacher("101","Rekha","Rosevilla",50000,"MCA","DS");

tobj[1] = new Teacher("102","Riya","Deepalayam",110000,"BBA","Commerce");

tobj[0].display();

tobj[1].display();

}

}

class Employees {

String Empid;

String Name;

String Address;

int Salary;

Employees(String id,String name,String addr,int salary){

this.Empid = id;

this.Name = name;

this.Address = addr;

this.Salary = salary;

}

void display(){

System.out.println("EmpID : " + this.Empid);

System.out.println("Name : " + this.Name);

System.out.println("Address : " + this.Address);

System.out.println("Salary : " + this.Salary);

}

}

class Teacher extends Employees{

String Department;

String Subject;

Teacher(String id,String name,String addr,int salary,String dept,String subj){

super(id,name,addr,salary);

this.Department=dept;

this.Subject=subj;

}

void display(){

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

super.display();

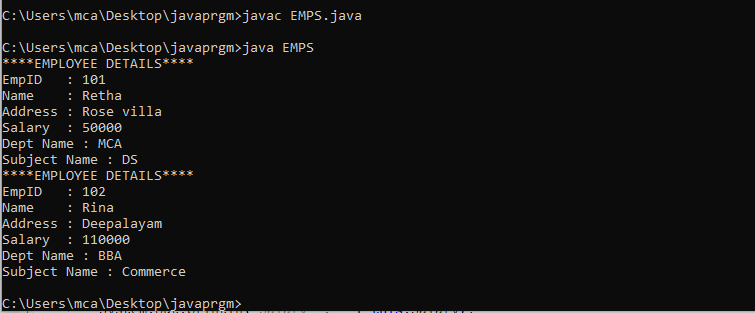
System.out.println("Dept Name : " + this.Department);

System.out.println("Subject Name : " + this.Subject);

}

}

**Output Screenshot**

****

**Object oriented programming lab**

**Name: sandra pm**

**Roll No:34**

**Batch: mca-b**

**Date:17-5-2022**

**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.

**Procedure**

import java.util.Scanner;

class Person {

    String name, gender, address;

    int age;

    public Person(String name, String gender, String address, int age) {

        this.name = name;

        this.gender = gender;

        this.address = address;

        this.age = age;

    }

}

class Employee extends Person {

    int empid;

    double salary;

    String company\_name, qualification;

    public Employee(String name, String gender, String address, int age, int empid, String company\_name,

            String qualification, double salary) {

        super(name, gender, address, age);

        this.empid = empid;

        this.company\_name = company\_name;

        this.qualification = qualification;

        this.salary = salary;

    }

}

class Teacher extends Employee {

    int teacher\_id;

    String department, subject;

    public Teacher(String name, String gender, String address, int age, int empid, String company\_name,

            String qualification, double salary, int teacher\_id, String department, String subject) {

        super(name, gender, address, age, empid, company\_name, qualification, salary);

        this.teacher\_id = teacher\_id;

        this.department = department;

        this.subject = subject;

    }

    void displayDetails(String emp) {

        System.out.println("The name of the " + emp + " is: " + this.name);

        System.out.println("The gender of the " + emp + " is: " + this.gender);

        System.out.println("The address of the " + emp + " is: " + this.address);

        System.out.println("The age of the " + emp + " is: " + this.age);

        System.out.println("The employee ID of the " + emp + " is: " + this.empid);

        System.out.println("The Company name of the " + emp + " is: " + this.company\_name);

        System.out.println("The qualification of the " + emp + " is: " + this.qualification);

        System.out.println("The salary of the " + emp + " is: " + this.salary);

        System.out.println("The teacher ID of the " + emp + " is: " + this.teacher\_id);

        System.out.println("The department of the " + emp + " is: " + this.department);

        System.out.println("The subject of the " + emp + " is: " + this.subject);

    }

}

class arrayMultiLevelInheritance {

    public static void main(String[] args) {

        int empnum;

        Scanner sc = new Scanner(System.in);

        System.out.print("Please enter the number of teacher employees you want: ");

        empnum = sc.nextInt();

        System.out.println("\n");

        Teacher[] teachers\_arr = new Teacher[empnum];

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

            String name, gender, address, company\_name, qualification, department, subject;

            int age, empid, teacher\_id;

            double salary;

            System.out.print("Enter the name of the " + (i + 1) + " teacher : ");

            name = sc.next();

            System.out.print("Enter the gender of the " + (i + 1) + " teacher : ");

            gender = sc.next();

            System.out.print("Enter the address of the " + (i + 1) + " teacher : ");

            address = sc.next();

            System.out.print("Enter the age of the " + (i + 1) + " teacher : ");

            age = sc.nextInt();

            System.out.print("Enter the emp ID of the " + (i + 1) + " teacher : ");

            empid = sc.nextInt();

            System.out.print("Enter the company name of the " + (i + 1) + " teacher : ");

            company\_name = sc.next();

            System.out.print("Enter the qualification of the " + (i + 1) + " teacher : ");

            qualification = sc.next();

            System.out.print("Enter the salary of the " + (i + 1) + " teacher : ");

            salary = sc.nextDouble();

            System.out.print("Enter the teacher ID of the " + (i + 1) + " teacher : ");

            teacher\_id = sc.nextInt();

            System.out.print("Enter the department of the " + (i + 1) + " teacher : ");

            department = sc.next();

            System.out.print("Enter the subject of the " + (i + 1) + " teacher : ");

            subject = sc.next();

            teachers\_arr[i] = new Teacher(name, gender, address, age, empid, company\_name,

                    qualification, salary, teacher\_id, department, subject);

            System.out.println("\n");

        }

        for (int i = 0; i < teachers\_arr.length; i++) {

            String txt = (i == 0) ? (i + 1) + "st"

                    : ((i == 1) ? (i + 1) + "nd" : ((i == 2) ? (i + 1) + "rd" : (i + 1) + "th"));

            teachers\_arr[i].displayDetails(txt);

            System.out.println("\n");

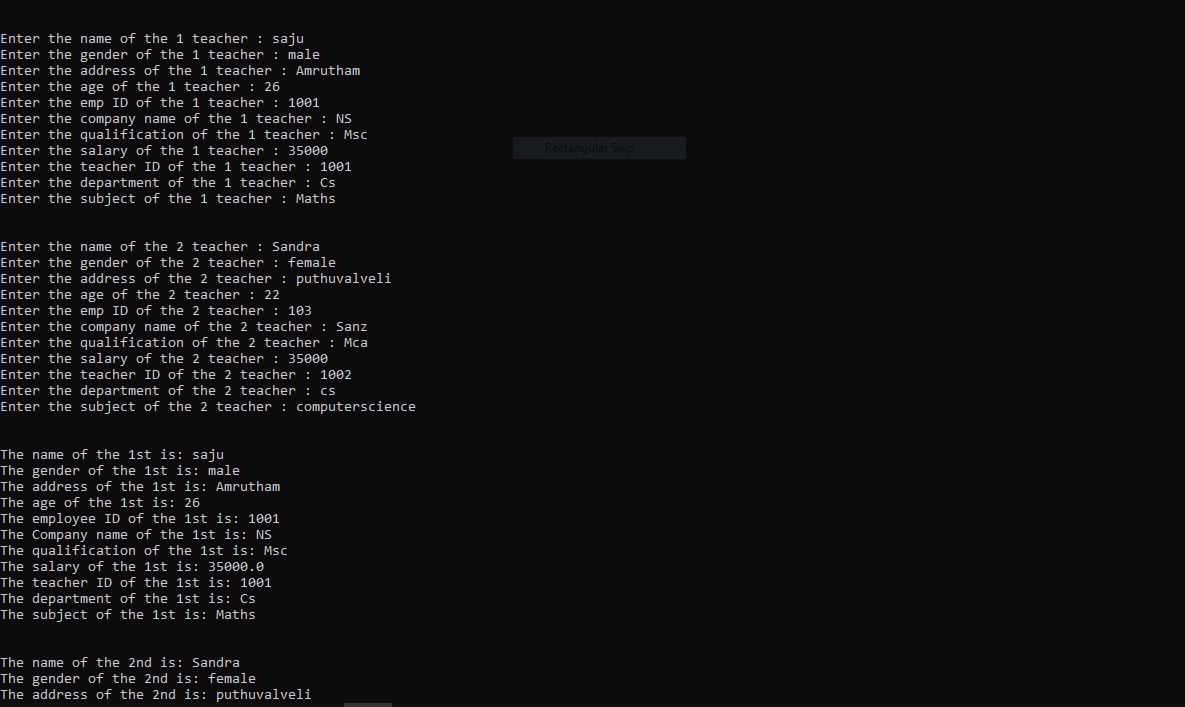
        }

        sc.close();

    }

}

**Output Screenshot**





**Object oriented programming lab**

**Name: sandra pm**

**Roll No:34**

**Batch: mca-b**

**Date:17-5-2022**

**Experiment No:13**

**Aim**

4. 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.

**Procedure**

import java.util.Scanner;

class Publisher

{ int p\_id;

String p\_name;

Publisher(int p\_id,String p\_name)

{ this.p\_id=p\_id;

this.p\_name=p\_name;

}

}

class Book extends Publisher

{ int b\_id;

String b\_name;

Book(int p\_id, String p\_name, int b\_id, String b\_name)

{ super(p\_id, p\_name);

this.b\_id=b\_id;

this.b\_name=b\_name;

}

}

class Literature extends Book

{ String cat;

Literature(int p\_id, String p\_name, int b\_id, String b\_name)

{ super(p\_id, p\_name, b\_id, b\_name);

this.cat="Literature";

}

void Display4()

{ System.out.println("\n\n.............Literature book details...........");

System.out.println("Category name : "+this.cat);

System.out.println("Publisher id : "+this.p\_id);

System.out.println("Publisher name : "+ this.p\_name);

System.out.println("Book id : " +this.b\_id);

System.out.println("Book name : "+ this.b\_name);

}

}

class Fiction extends Book

{ String cat;

Fiction(int p\_id, String p\_name, int b\_id, String b\_name)

{ super(p\_id, p\_name, b\_id, b\_name);

this.cat="Fiction";

}

void Display4()

{ System.out.println("\n\n.............Fiction book details...........");

System.out.println("Category name : "+this.cat);

System.out.println("Publisher id : "+this.p\_id);

System.out.println("Publisher name : "+ this.p\_name);

System.out.println("Book id : " +this.b\_id);

System.out.println("Book name : "+ this.b\_name);

}

}

public class PublisherBooks

{ public static void main(String[] args)

{ int p\_id, b\_id;

String p\_name, b\_name,t;

Scanner s=new Scanner(System.in);

System.out.print("\nEnter the type of book(Type 'l/L' for Literature/'f/F' for Fiction)? ");

t=s.next();

if(t.equals("l")||t.equals("L"))

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

p\_id=s.nextInt();

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

p\_name=s.next();

System.out.print("Enter the Book ID :");

b\_id=s.nextInt();

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

b\_name=s.next();

Literature lit=new Literature(p\_id, p\_name, b\_id, b\_name);

lit.Display4();

}

else if(t.equals("f")||t.equals("F"))

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

p\_id=s.nextInt();

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

p\_name=s.next();

System.out.print("Enter the Book ID :");

b\_id=s.nextInt();

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

b\_name=s.next();

Fiction fic=new Fiction(p\_id, p\_name, b\_id, b\_name);

fic.Display4();

}

else

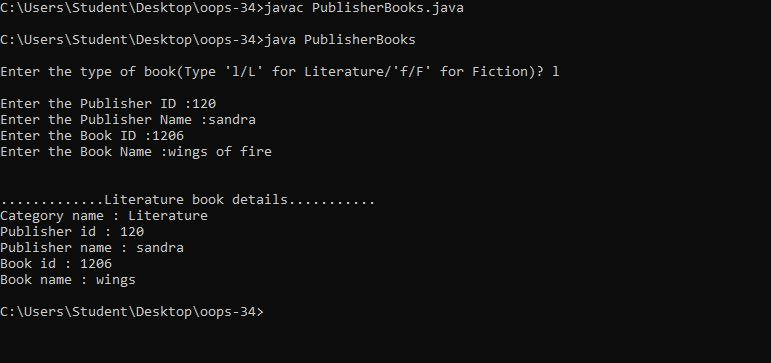
{ System.out.println("\n\n!!!!!!!!!!!!!!!Entry for type of book is not valid!!!!!!!!!!!!!! ");

}

}

}

**Output**

****

**Name: sandra pm**

**Roll No:34**

**Batch: mca-b**

**Date:17-5-2022**

**Object oriented programming lab**

**Experiment No:14**

**Aim:**

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

**Procedure**

import java.util.Scanner;

class sports{

String sport;

int Rating;

sports(String spo, int ra){

sport = spo;

Rating = ra;

}

}

class student extends sports{

String Grade;

double Overall\_per;

student(String spo, int ra,String gd, double per ){

super(spo, ra);

Grade = gd;

Overall\_per = per;

}

}

public class result extends student {

result(String spo, int ra,String gd, double per ){

super(spo, ra, gd, per);

}

void display(){

System.out.println("\nSports Details of Student");

System.out.println("Sport :"+sport);

System.out.println("Rating :"+Rating);

System.out.println("\nAcademic Details of Student");

System.out.println("Academic Grade :"+Grade);

System.out.println("Overall percentage :"+Overall\_per);

}

public static void main(String[] args) {

Scanner sc =new Scanner(System.in);

System.out.println("\nEnter the Sports Details of Student");

System.out.println("\n Sport: ");

String a =sc.next();

System.out.println("\n Sport Rating out of 10: ");

int b =sc.nextInt();

System.out.println("\nEnter the Sports Details of Student");

System.out.println("\n Academic Grade: ");

String c =sc.next();

System.out.println("\n Overall percentage: ");

double d =sc.nextDouble();

sc.close();

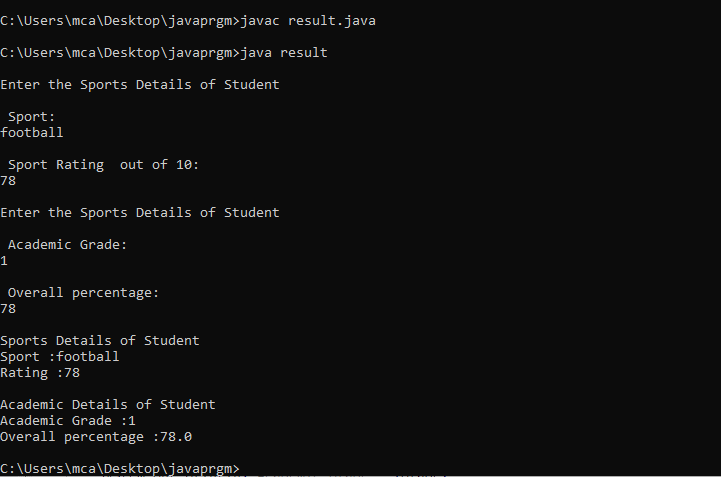
result obj= new result(a,b,c,d);

obj.display();

}

}

**Output Screenshot**

****

**Name: sandra pm**

**Roll No:34**

**Batch :S2 MCA**

**Date:24/05/2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**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.

**Procedure**

import java.util.Scanner;

interface Circlerect{

void area();

void perimeter();

}

class Circle implements Circlerect{

int r;

double pi = 3.14, area,perimeter;

public void area(){

Scanner s = new Scanner(System.in);

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

r = s.nextInt();

area = pi \* r \* r;

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

}

public void perimeter(){

perimeter = 2 \* pi \* r;

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

}

}

class Rectangle implements Circlerect{

int l,b,area,perimeter;

public void area(){

Scanner s = new Scanner(System.in);

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

l=s.nextInt();

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

b=s.nextInt();

area=l\*b;

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

}

public void perimeter(){

perimeter=2 \* (l + b);

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

}

}

public class Interfacecirclerect{

public static void main(String args []){

int n;

while(true){

Circlerect c=new Circle();

Circlerect r=new Rectangle();

System.out.println("\n\nchoose the operations you can do:");

System.out.println("1.circle\n\n 2.Rectangle\n\n3.exit");

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

Scanner s = new Scanner(System.in);

n=s.nextInt();

switch(n)

{

case 1:System.out.println("circle");

c.area();

c.perimeter();

break;

case 2:System.out.println("Rectangle");

r.area();

r.perimeter();

break;

case 3:System.exit(0);

break;

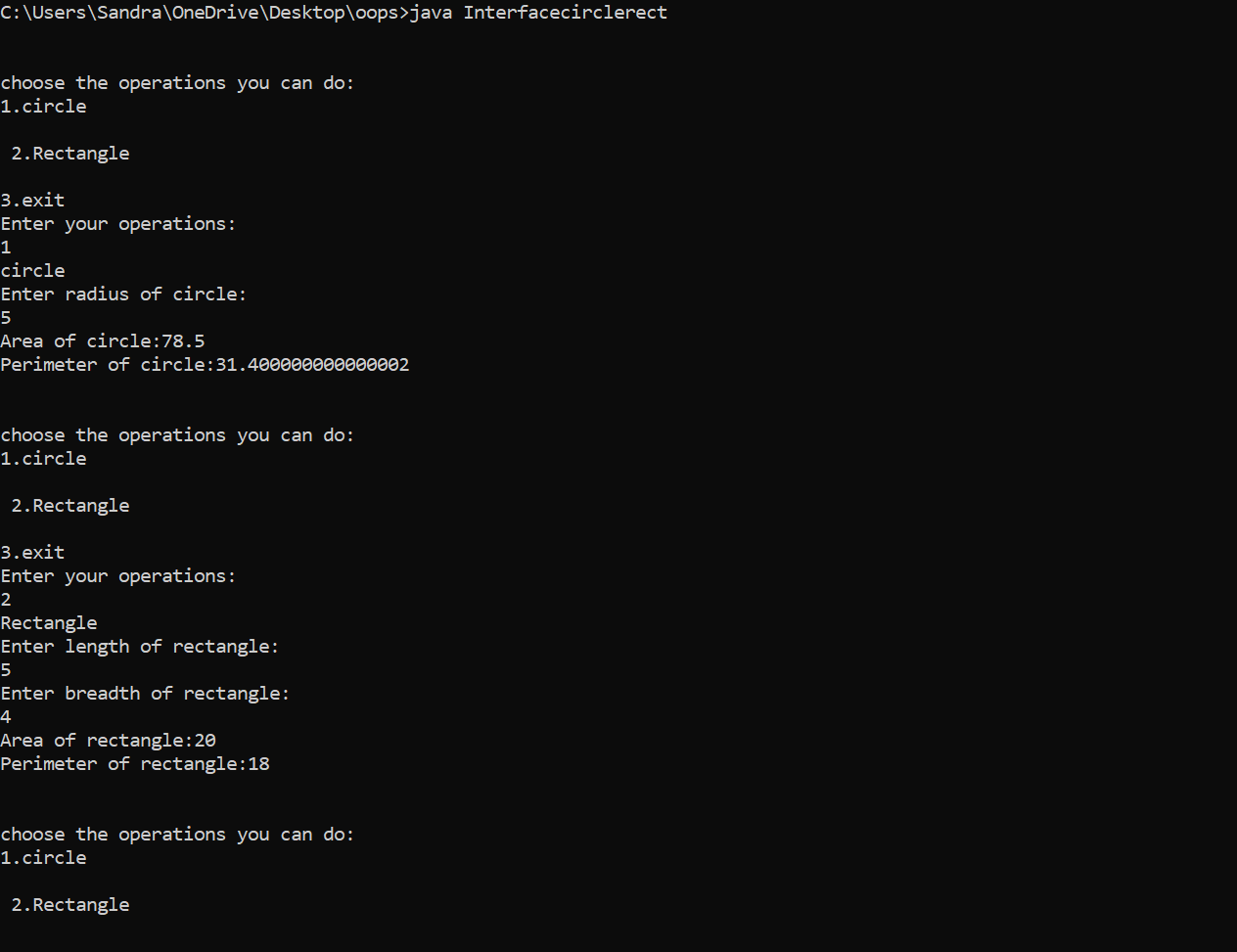
}

}

}

}

**Output**



**OBJECT ORIENTED PROGRAMMING LAB**

**Name: sandra pm**

**Roll No:34**

**Batch: S2 MCA B**

**Date:24/05/2022**

**Experiment No.: 16**

**Aim**

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

**Procedure**

import java.util.Scanner;

interface calc{

void calculate();

}

class bill implements calc{

String date,name,p\_id;

int quantity;

double unit\_price,total,namount=0;

Scanner sc = new Scanner(System.in);

public void getdata(){

System.out.println("\nEnter product id:");

p\_id = sc.nextLine();

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

name = sc.nextLine();

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

quantity = sc.nextInt();

System.out.println("Enter the unit price:");

unit\_price = sc.nextDouble();

}

public void calculate(){

total = quantity \* unit\_price;

}

public void display(){

System.out.println(p\_id+"\t\t"+name+"\t\t"+quantity+"\t\t"+unit\_price+"\t"+total);

}

}

public class BillCalc {

public static void main(String[] args) {

int n,i;

double namount=0,t;

int ran;

String date;

t = Math.random() \*1000000;

ran = (int) t;

Scanner sc = new Scanner(System.in);

System.out.println("Order no. #"+ran);

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

date = sc.nextLine();

System.out.println("Enter how many products are there:");

n = sc.nextInt();

bill ob[] = new bill[n];

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

ob[i] = new bill();

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

ob[i].getdata();

ob[i].calculate();

}

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

System.out.println("Product Id Name Quantity unit price Total ");

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

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

ob[i].display();

namount += ob[i].total;

}

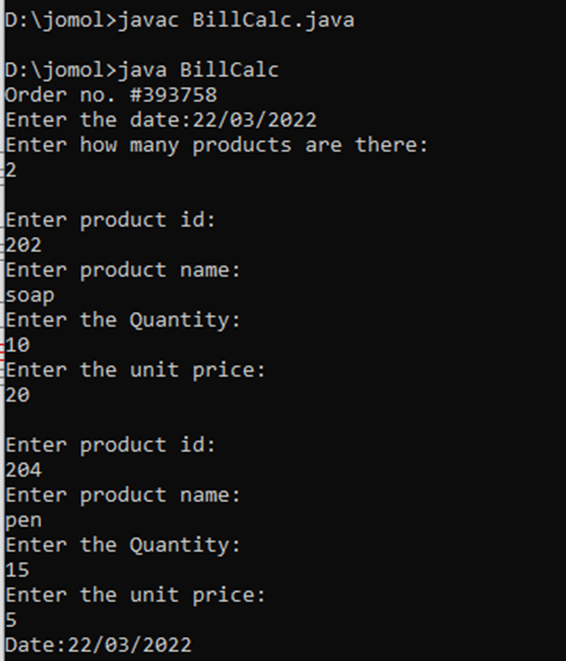
System.out.println("--------------------------------------------------------------");

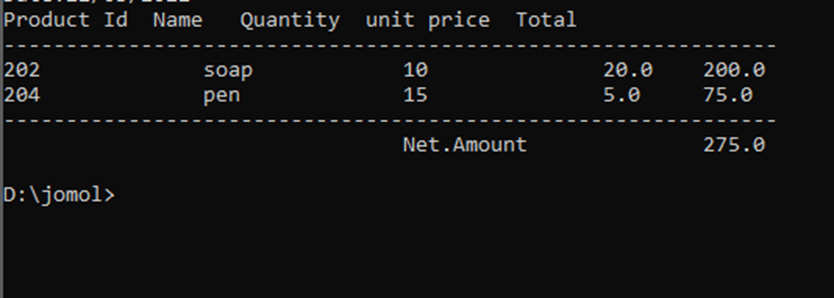
System.out.println("\t\t\t\tNet.Amount\t\t"+ namount);

}

}

Output

****

****

**Name: Sandra P M**

**Roll No:34**

**Batch: MCA B**

**Date:31-05-2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**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**.**

**AreaCalculation.java**

import java.util.\*;

import graphics.\*;

public class AreaCalculation {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

int choice,isexit=0;

while(isexit==0){

double length, breadth, side, radius;

System.out.println("\n1. Area of Triangle.\n2. Area of Circle.\n3. Area of Square.\n4. Area of Rectangle.\n5. Exit");

System.out.print("Please enter the operation choice to perform - ");

choice= sc.nextInt();

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

switch(choice){

case 1:{

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

length= sc.nextDouble();

System.out.print("Enter the height of the triangle : ");

breadth= sc.nextDouble();

Triangle triangle = new Triangle(length, breadth);

triangle.Area();

break;

}

case 2:{

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

radius= sc.nextDouble();

Circle cir= new Circle(radius);

cir.Area();

break;

}

case 3:{

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

side= sc.nextDouble();

Square square= new Square(side);

square.Area();

break;

}

case 4:{

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

length= sc.nextDouble();

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

breadth= sc.nextDouble();

Rectangle rec= new Rectangle(length, breadth);

rec.Area();

break;

}

case 5:{

isexit=1;

break;

}

default:{

break;

}

}

}

sc.close();

}

}

Triangle.java

package graphics;

interface AreaInterface {

void Area();

}

public class Triangle implements AreaInterface{

double length, breadth;

public Triangle(double length, double breadth){

this.length= length;

this.breadth= breadth;

}

public void Area() {

double area= 0.5 \* this.length \* this.breadth;

System.out.println("The area of the given triangle is : "+area);

}

}

Rectangle.java

package graphics;

interface AreaInterface {

public void Area();

}

public class Rectangle implements AreaInterface{

double length, breadth;

public Rectangle(double length, double breadth){

this.length= length;

this.breadth= breadth;

}

public void Area() {

double area= 0.5 \* this.length \* this.breadth;

System.out.println("The area of the given rectangle is : "+area);

}

}

Circle.java

package graphics;

interface AreaInterface {

void Area();

}

public class Circle implements AreaInterface{

double radius;

public Circle(double radius){

this.radius= radius;

}

public void Area() {

double area= 3.14 \* this.radius \* this.radius;

System.out.println("The area of the given circle is : "+area);

}

}

Square.java

package graphics;

interface AreaInterface {

void Area();

}

public class Square implements AreaInterface{

double side;

public Square(double side){

this.side= side;

}

public void Area() {

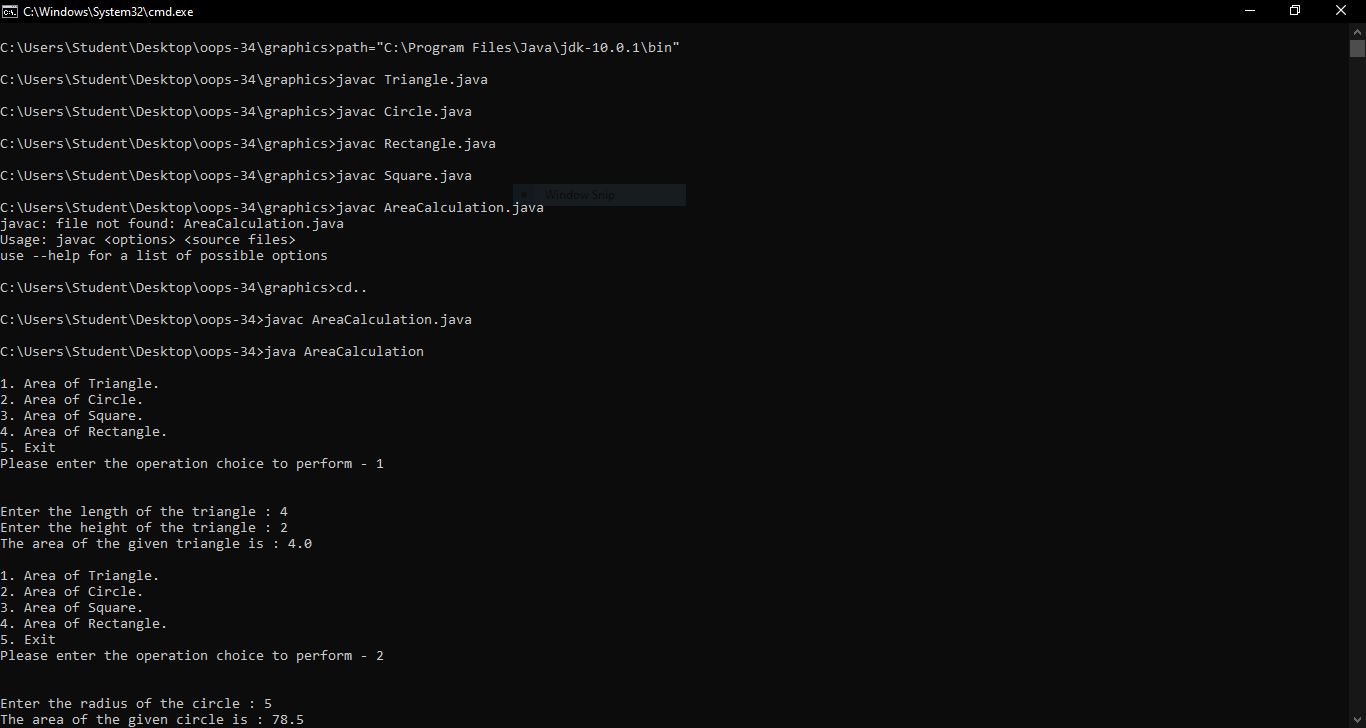
double area= this.side \* this.side;

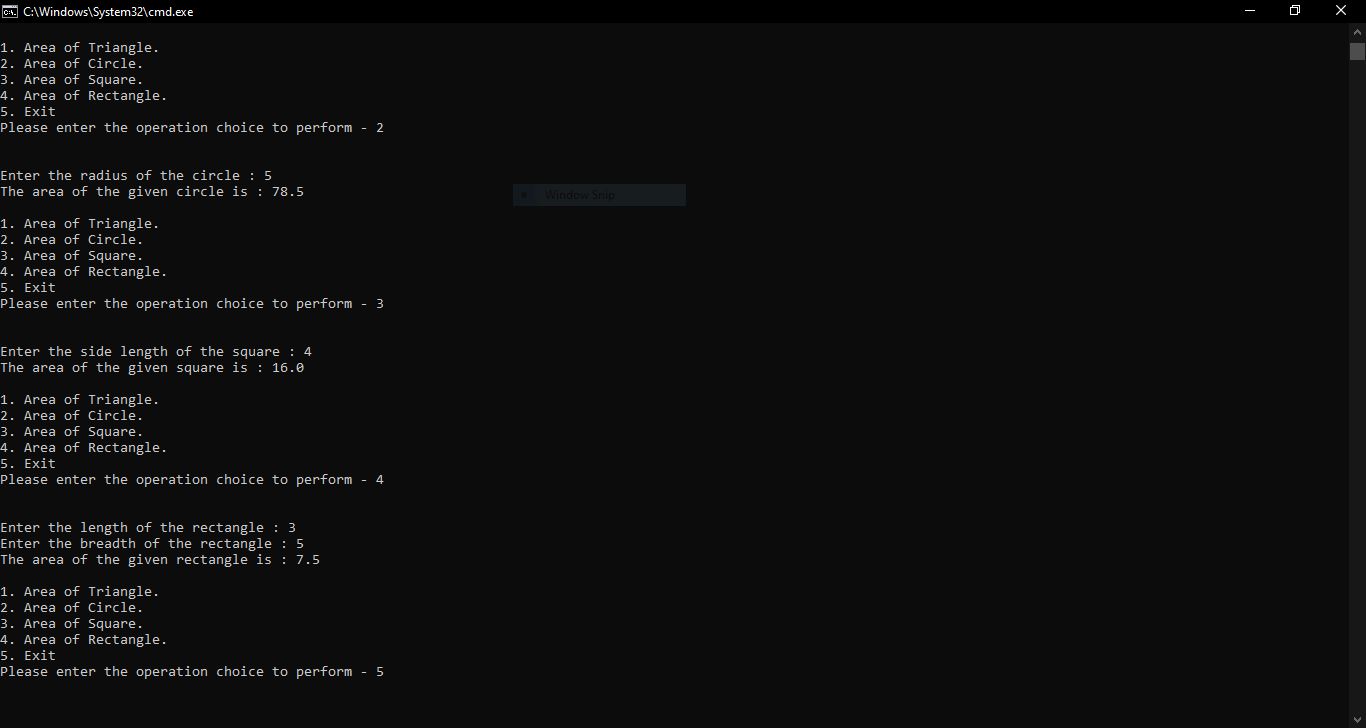
System.out.println("The area of the given square is : "+area);

}

}

**Output**

****

****

**Name: Sandra P M**

**Roll No:34**

**Batch: MCA B**

**Date:31-05-2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**Experimentno:-18**

**AIM**

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

**Procedure**

public class CustomExceptionExample {

public static class InvalidUserException extends Exception {

public InvalidUserException() {

super("Invalid username / password provided!");

}

}

public static void main(String[] args) {

String username = "san";

String password = "pass";

try {

if (username.equals("user") && password.equals("pass")) {

System.out.println("Authenticated successfully!");

} else {

throw new InvalidUserException();

}

} catch (InvalidUserException e) {

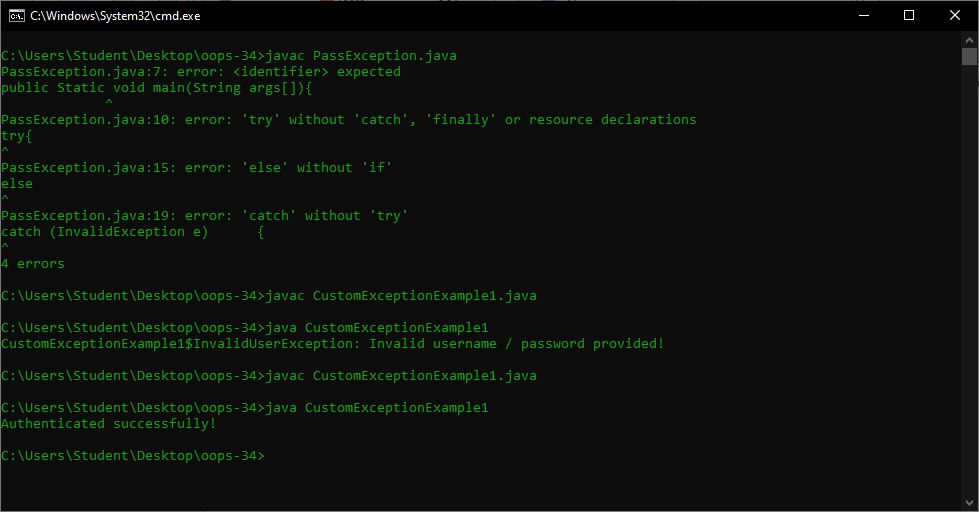
System.out.println(e);

}

}

}

**Output**



**OBJECT ORIENTED PROGRAMMING LAB**

**Name: Sandra P M**

**Roll No:34**

**Batch: MCA B**

**Date:31-05-2022**

**Experiment no:19**

**AIM**

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

**Procedure**

import java.util.\*;

class MyException extends Exception {

public MyException(String value) {

super(value);

}

}

class Main {

public static void main(String args[]) {

int totalNums;

int i;

int temp, count = 0;

int sum = 0;

Scanner sc = new Scanner(System.in);

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

totalNums = Integer.parseInt(sc.nextLine());

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

try {

temp = Integer.parseInt(sc.nextLine());

if (temp > 0) {

sum += temp;

count += 1;

} else {

throw new MyException(Integer.toString(temp));

}

} catch (MyException ex) {

System.out.print(ex.getMessage());

System.out.println(" - Not a positive number");

}

}

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

System.out.println(count);

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

System.out.println(sum);

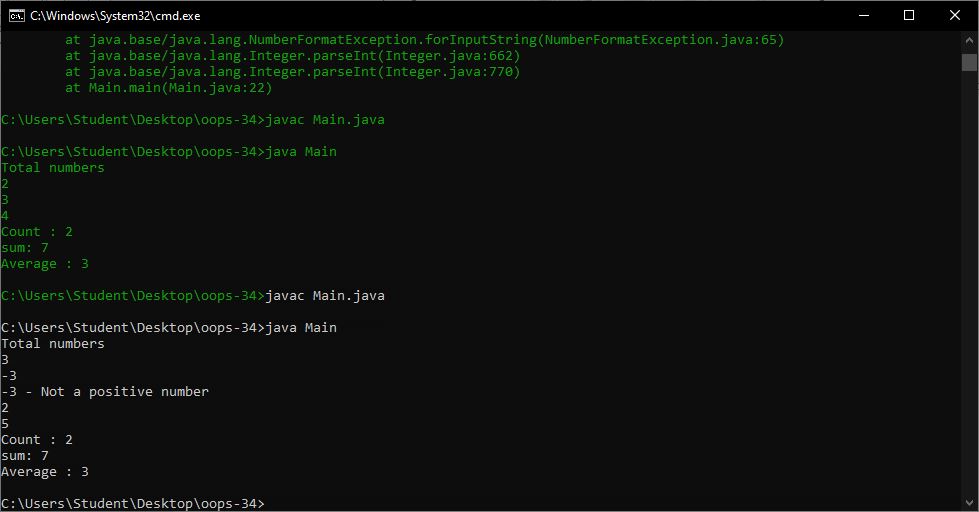
System.out.print("Average : ");

System.out.println(sum / count);

}

}

**Output**

****

**OBJECT ORIENTED PROGRAMMING LAB**

**Name: Sandra P M**

**Roll No:34**

**Batch: MCA B**

**Date:31-05-2022**

**Experiment no:20**

**AIM**

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

import java.util.\*;

class fibonacci implements Runnable {

int l;

fibonacci(int n) {

l = n;

}

public void run() {

int c;

int a = 0, b = 1;

System.out.print(a + " " + b);

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

c = a + b;

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

a = b;

b = c;

}

}

}

class even implements Runnable {

int l;

even(int n) {

l = n;

}

public void run() {

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

if (i % 2 == 0)

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

}

System.out.println("");

}

}

class My{

public static void main(String args[]) {

Scanner sc = new Scanner(System.in);

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

int l = sc.nextInt();

fibonacci f = new fibonacci(l);

Thread T1 = new Thread(f);

T1.start();

even e = new even(l);

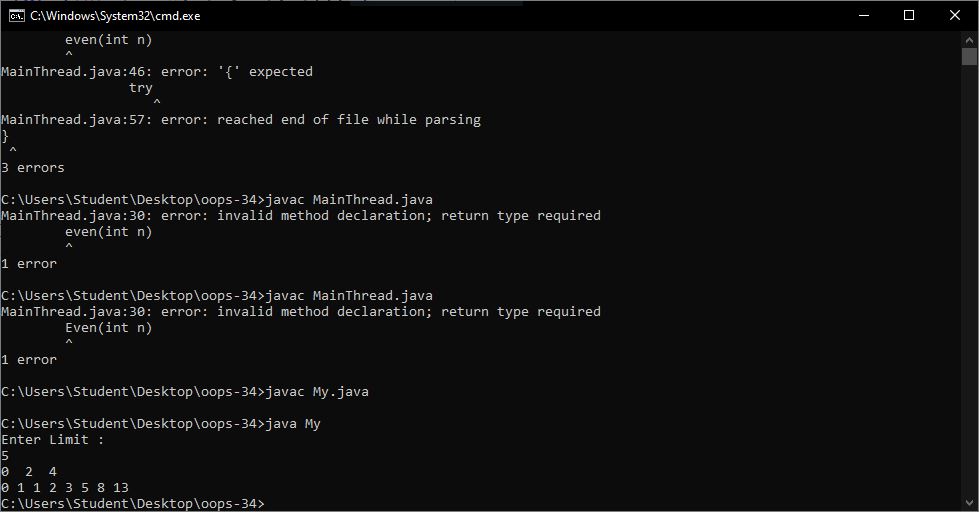
Thread T2 = new Thread(e);

T2.start();

}

}

**Output**

****

**Name: Sandra P M**

**Roll No:34**

**Batch: MCA B**

**Date:31-05-2022**

**Experiment no:21**

**Aim**

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

**Procedure**

class Stack {

private int arr[];

private int top;

private int capacity;

Stack(int size) {

arr = new int[size];

capacity = size;

top = -1;

}

public void push(int x) {

if (isFull()) {

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

System.exit(1);

}

System.out.println("Inserting " + x);

arr[++top] = x;

}

public int pop() {

if (isEmpty()) {

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

System.exit(1);

}

return arr[top--];

}

public int getSize() {

return top + 1;

}

public Boolean isEmpty() {

return top == -1;

}

public Boolean isFull() {

return top == capacity - 1;

}

public void printStack() {

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

System.out.print(arr[i] + ", ");

}

}

public static void main(String[] args) {

Stack stack = new Stack(5);

stack.push(1);

stack.push(2);

stack.push(3);

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

stack.printStack();

stack.pop();

System.out.println("\nAfter popping out");

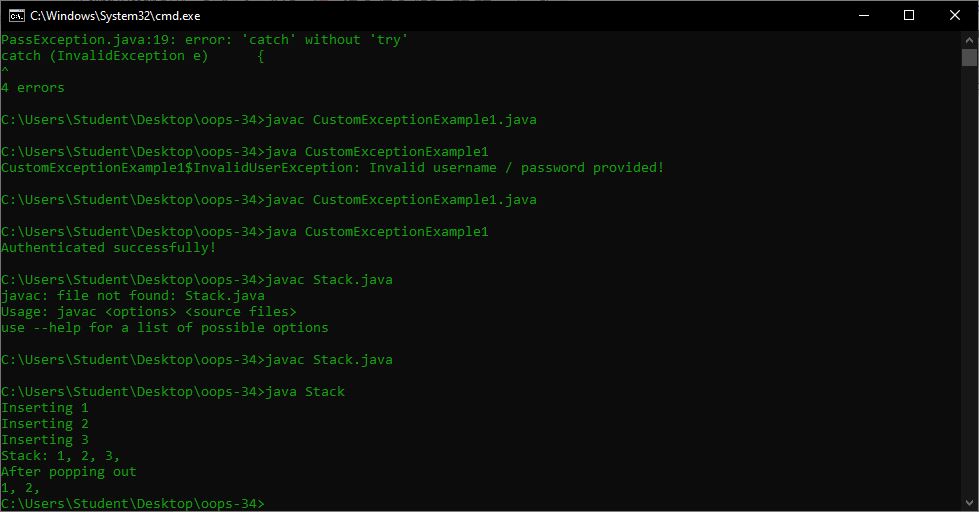
stack.printStack();

}

}

}

**Output**

****

**Experiment no:22**

**AIM**

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

**Procedure**

import java.util.ArrayList;

import java.util.Collections;

class Mycls {

public static void main(String[] args) {

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

data.add("A");

data.add("B");

data.add("C");

data.add("D");

data.set(1, "BB");

System.out.println(data);

System.out.println(data.get(0));

System.out.println(data.get(1));

data.remove(0);

System.out.println(data);

System.out.println(data.size());

for (String d : data) {

System.out.println(d);

}

Collections.sort(data);

System.out.println(data);

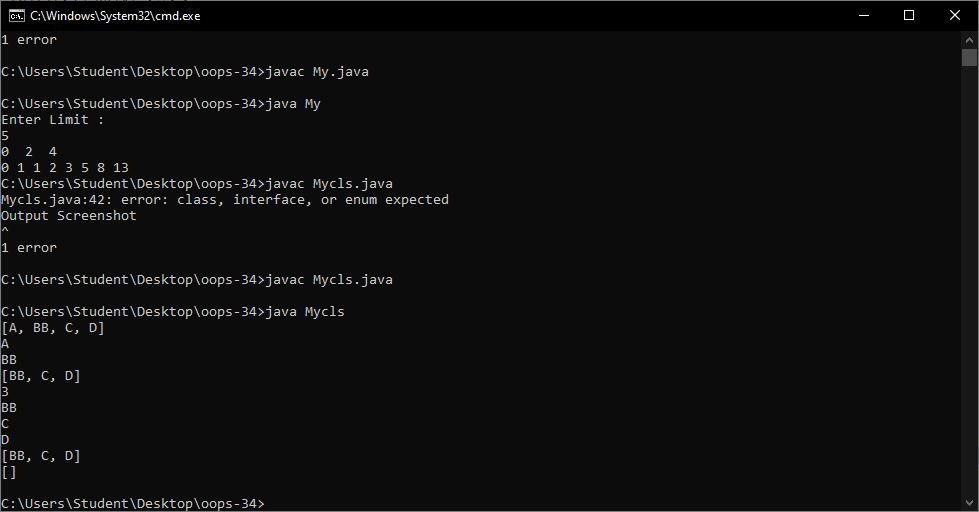
data.clear();

System.out.println(data);

}

}

**Output Screenshot**

****

**Program to demonstrate the creation of queue object using the Priority Queue class.**

import java.util.\*;

class Collection\_Framework\_Queue {

public static void main(String args[]) {

Queue<Integer> q = new PriorityQueue<Integer>(new Comp());

int ch;

Scanner sc = new Scanner(System.in);

do {

System.out.println("\n1.ADD\n2.PEEK\n3.POLL or REMOVE\n4.DISPLAY\n5.EXIT");

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

ch = sc.nextInt();

switch (ch) {

case 1:

System.out.println("\n\tEnter Integer : ");

int n1 = sc.nextInt();

q.add(n1);

System.out.println("\n\tADDED SUCCESSFULLY ! ! ! ");

break;

case 2:

if (q.isEmpty()) {

System.out.print("\n\tQueue Empty ! ! !");

} else {

System.out.print("\n\tPeeked element is " + q.peek());

}

break;

case 3:

if (!q.isEmpty()) {

System.out.print("\n\tRemoved element is " + q.poll());

} else {

System.out.print("\n\tQueue Empty ! ! !");

}

break;

case 4:

if (!q.isEmpty()) {

System.out.print("\nSize of queue : " + q.size());

System.out.print("\nQueue elements : " + q);

System.out.println("\nQueue elements are");

for (int i : q) {

System.out.println(i);

}

} else {

System.out.print("\n\tQueue Empty ! ! !");

}

break;

case 5:

break;

default:

System.out.println("\n\tPlease enter valid choice ! ! ! ");

}

} while (ch != 5);

}

}

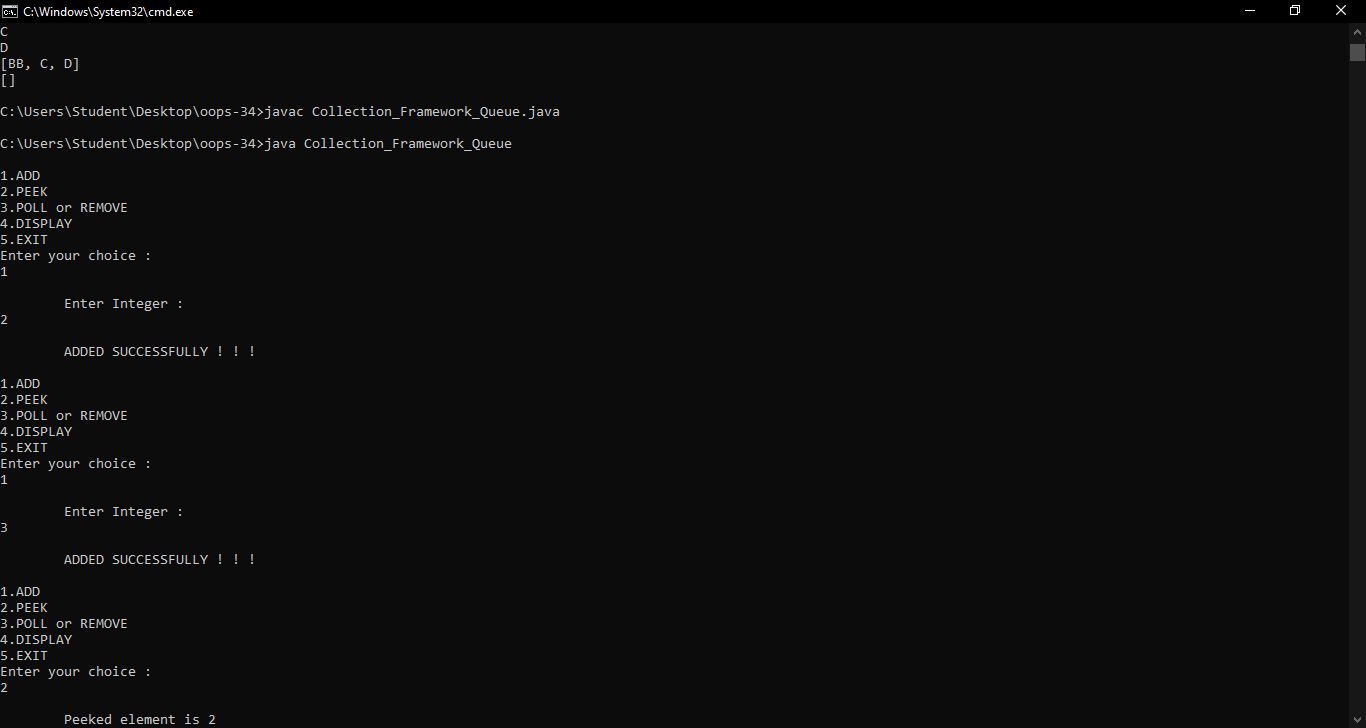
class Comp implements Comparator<Integer> {

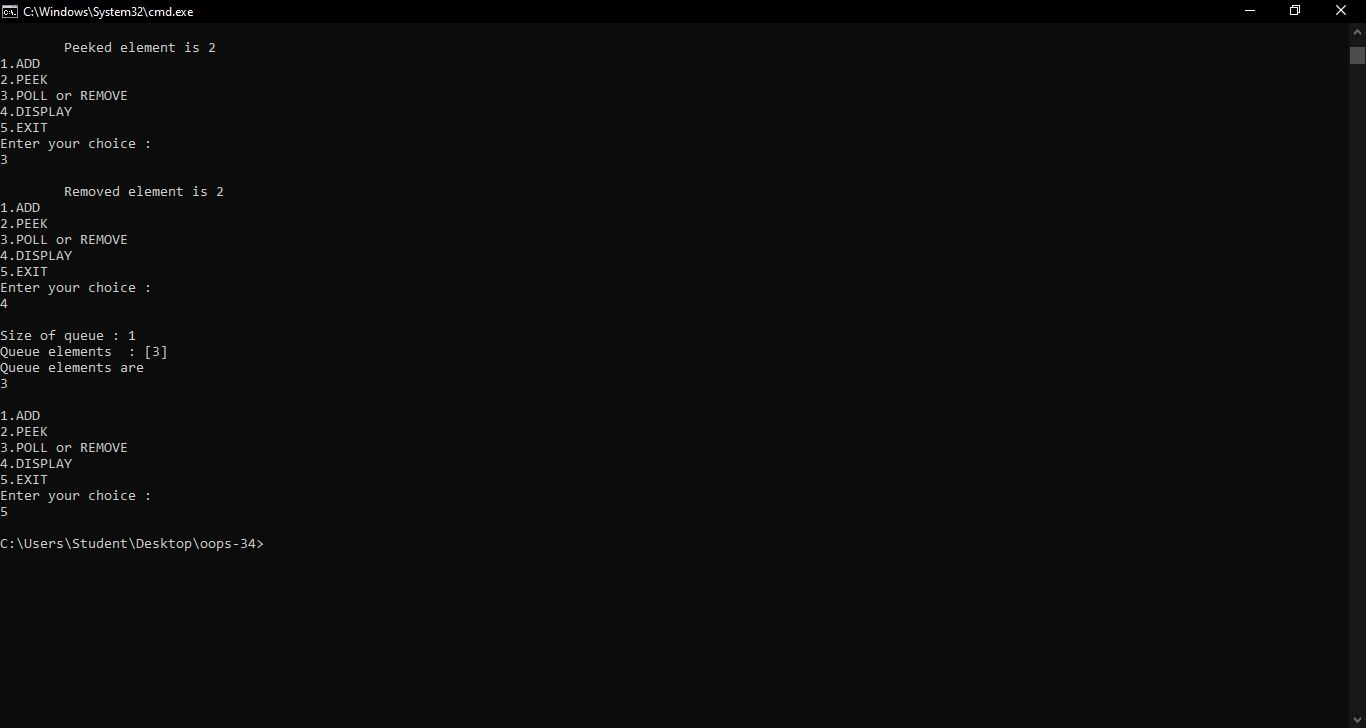
public int compare(Integer a, Integer b) {

return a % 10 > b % 10 ? 1 : -1;

}

}

**Output Screenshot**

****

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

import java.util.\*;

class deque

{

public static void main(String[] args)

{

Deque<String> deque = new LinkedList<String>();

deque.add("Element 1 (Tail)");

deque.addFirst("Element 2 (Head)");

deque.addLast("Element 3 (Tail)");

deque.push("Element 4 (Head)");

deque.offer("Element 5 (Tail)");

deque.offerFirst("Element 6 (Head)");

System.out.println(deque + "\n");

deque.removeFirst();

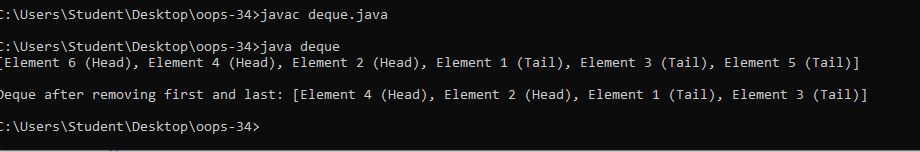
deque.removeLast();

System.out.println("Deque after removing " + "first and last: " + deque);

}

}

**Output**

****

***Write a Java program to compare two hash set***

import java.util.\*;

public class Exercise10 {

public static void main(String[] args) {

HashSet<String> h\_set = new HashSet<String>();

h\_set.add("Red");

h\_set.add("Green");

h\_set.add("Black");

h\_set.add("White");

HashSet<String>h\_set2 = new HashSet<String>();

h\_set2.add("Red");

h\_set2.add("Pink");

h\_set2.add("Black");

h\_set2.add("Orange");

HashSet<String>result\_set = new HashSet<String>();

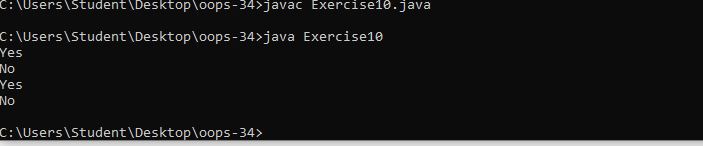
for (String element : h\_set){

System.out.println(h\_set2.contains(element) ? "Yes" : "No");

}

}

}



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

import java.util.\*;

class HashMapDemo {

public static void main(String args[]) {

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

hm.put("Anu", new Integer(1));

hm.put("sinu", new Integer(2));

hm.put("Jinu", new Integer(3));

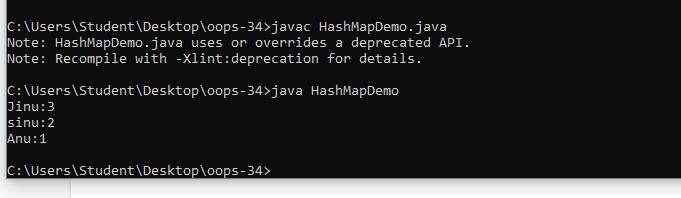
for (Map.Entry<String, Integer> me : hm.entrySet()) {

System.out.print(me.getKey() + ":");

System.out.println(me.getValue()); }

} }

**output**

****

**Name: Sandra P M**

**Roll No:34**

**Batch: MCA B**

**Date:09-06-2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: co5**

**Aim**

**procedure**

import java.awt.\*;

import java.awt.event.\*;

public class calculator implements ActionListener

{

int c,n;

String s1,s2,s3,s4,s5;

Frame f;

Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;

Panel p;

TextField tf;

GridLayout g;

calculator()

{

f = new Frame("My calculator");

p = new Panel();

f.setLayout(new FlowLayout());

b1 = new Button("0");

b1.addActionListener(this);

b2 = new Button("1");

b2.addActionListener(this);

b3 = new Button("2");

b3.addActionListener(this);

b4 = new Button("3");

b4.addActionListener(this);

b5 = new Button("4");

b5.addActionListener(this);

b6 = new Button("5");

b6.addActionListener(this);

b7 = new Button("6");

b7.addActionListener(this);

b8 = new Button("7");

b8.addActionListener(this);

b9 = new Button("8");

b9.addActionListener(this);

b10 = new Button("9");

b10.addActionListener(this);

b11 = new Button("+");

b11.addActionListener(this);

b12 = new Button("-");

b12.addActionListener(this);

b13 = new Button("\*");

b13.addActionListener(this);

b14 = new Button("/");

b14.addActionListener(this);

b15 = new Button("%");

b15.addActionListener(this);

b16 = new Button("=");

b16.addActionListener(this);

b17 = new Button("C");

b17.addActionListener(this);

tf = new TextField(20);

f.add(tf);

g = new GridLayout(4,4,10,20);

p.setLayout(g);

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

p.add(b10);p.add(b11);p.add(b12);p.add(b13);p.add(b14);p.add(b15);p.add(b16);p.add(b17);

f.add(p);

f.setSize(300,300);

f.setVisible(true);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==b1)

{

s3 = tf.getText();

s4 = "0";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b2)

{

s3 = tf.getText();

s4 = "1";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b3)

{

s3 = tf.getText();

s4 = "2";

s5 = s3+s4;

tf.setText(s5);

}if(e.getSource()==b4)

{

s3 = tf.getText();

s4 = "3";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b5)

{

s3 = tf.getText();

s4 = "4";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b6)

{

s3 = tf.getText();

s4 = "5";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b7)

{

s3 = tf.getText();

s4 = "6";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b8)

{

s3 = tf.getText();

s4 = "7";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b9)

{

s3 = tf.getText();

s4 = "8";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b10)

{

s3 = tf.getText();

s4 = "9";

s5 = s3+s4;

tf.setText(s5);

}

if(e.getSource()==b11)

{

s1 = tf.getText();

tf.setText("");

c=1;

}

if(e.getSource()==b12)

{

s1 = tf.getText();

tf.setText("");

c=2;

}

if(e.getSource()==b13)

{

s1 = tf.getText();

tf.setText("");

c=3;

}

if(e.getSource()==b14)

{

s1 = tf.getText();

tf.setText("");

c=4;

}

if(e.getSource()==b15)

{

s1 = tf.getText();

tf.setText("");

c=5;

}

if(e.getSource()==b16)

{

s2 = tf.getText();

if(c==1)

{

n = Integer.parseInt(s1)+Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

else

if(c==2)

{

n = Integer.parseInt(s1)-Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

else

if(c==3)

{

n = Integer.parseInt(s1)\*Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

if(c==4)

{

try

{

int p=Integer.parseInt(s2);

if(p!=0)

{

n = Integer.parseInt(s1)/Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

else

tf.setText("infinite");

}

catch(Exception i){}

}

if(c==5)

{

n = Integer.parseInt(s1)%Integer.parseInt(s2);

tf.setText(String.valueOf(n));

}

}

if(e.getSource()==b17)

{

tf.setText("");

}

}

public static void main(String[] abc)

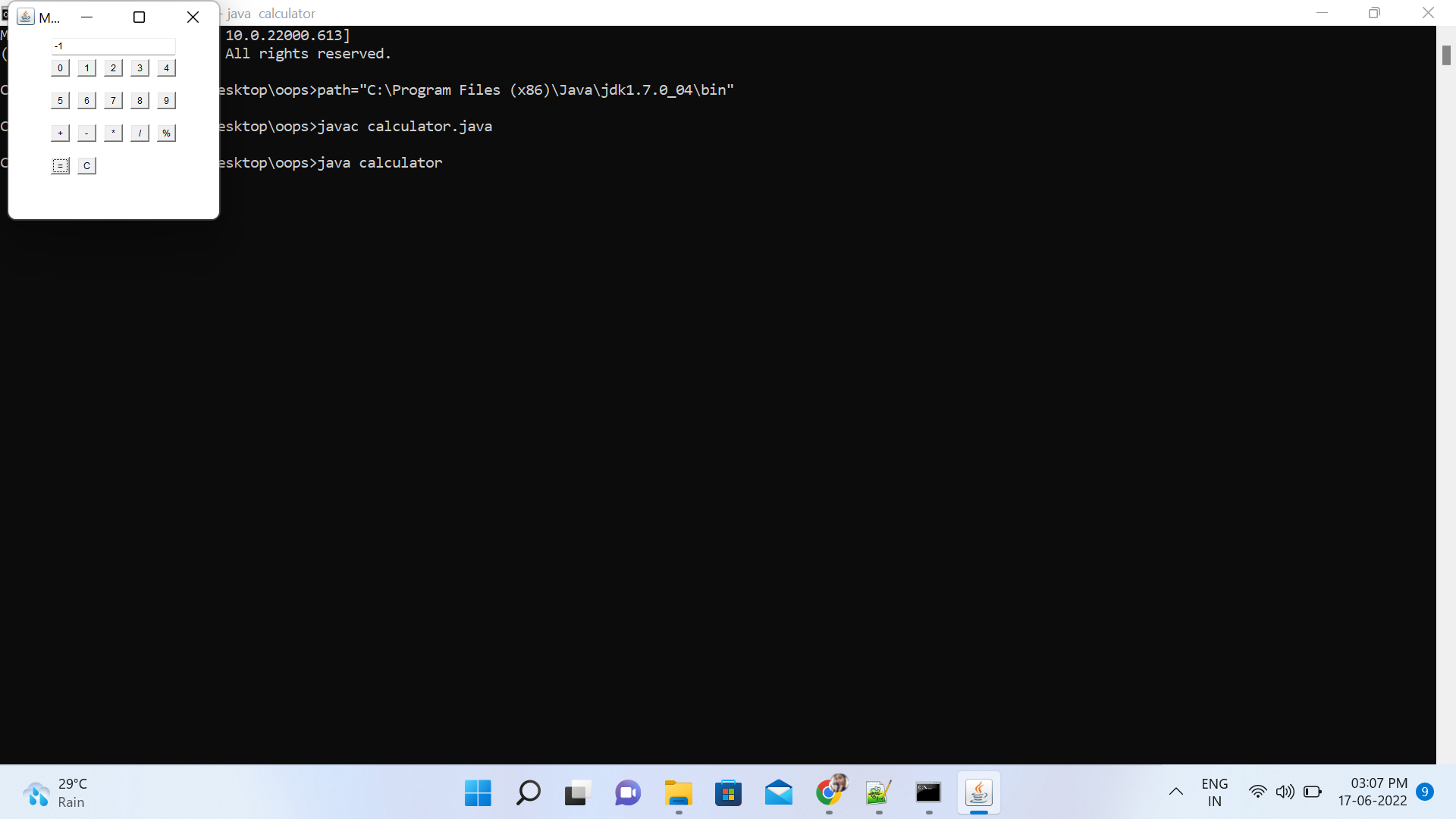
{

calculator v = new calculator();

}

}

output



**Name: Sandra PM**

**Roll No:34**

**Batch:MCA**

**Date:30/05/22**

**Object oriented programming lab**

**Experiment No.:**

**Aim**

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

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.\*;

import java.util.\*;

import java.io.File;

class read {

public static void main(String[] args) {

String var = "";

Scanner scan = new Scanner(System.in);

System.out.println("Enter the text to create file : type ENTER key 3 times to stop");

while (!var.endsWith("\n\n\n"))

var = var + scan.nextLine() + "\n";

try {

File file = new File("file4.txt");

FileWriter fw = new FileWriter(file);

fw.write(var);

fw.close();

System.out.println("Reading File content");

FileReader fr = new FileReader("file4.txt");

String str = "";

int i;

while ((i = fr.read()) != -1) {

str += (char) i;

}

System.out.println(str);

fr.close();

} catch (IOException e) {

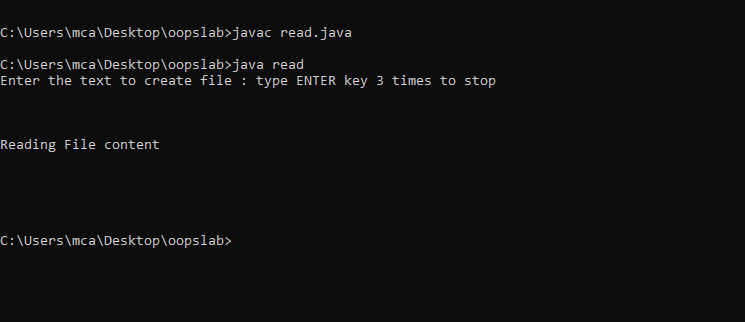
System.out.println("There are some exception");

}

}

}

**Output**

****

**Write a program to copy one file to another.**

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.\*;

import java.util.\*;

import java.io.File;

public class cp{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

System.out.println("enter the file name");

String source=sc.nextLine();

try {

FileReader fr=new FileReader(source);

String str = "";

int i;

System.out.println("Reading from file "+source);

while ((i = fr.read()) != -1) {

str += (char) i;

}

System.out.println(str);

System.out.println("\n Enter the filename to copy");

String destination=sc.nextLine();

File file=new File(destination);

FileWriter fw = new FileWriter(file);

fw.write(str);

fr.close();

fw.close();

System.out.println("Copied from "+source+" to "+destination+ " Successfully..!");

} catch (Exception e) {

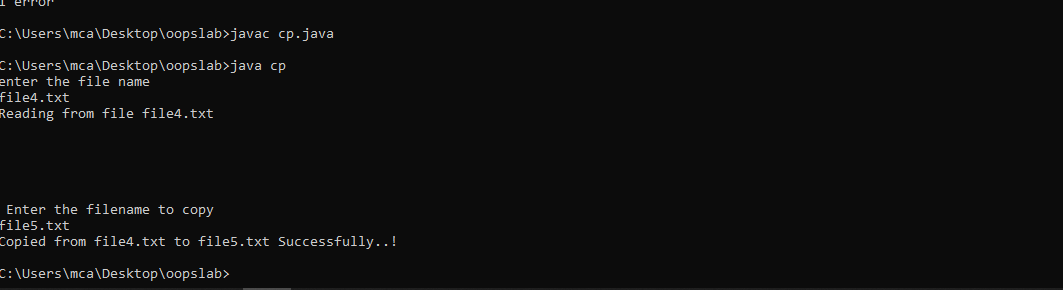
System.out.println("Exception Occured");

}

}

}

**Output**

****

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

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.\*;

import java.util.\*;

import java.io.File;

public class oddeven {

public static void main(String[] args) {

try {

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

BufferedReader br = new BufferedReader(fr);

File file1 = new File("oddnumbers.txt");

FileWriter fw1 = new FileWriter(file1);

File file2 = new File("evennumbers.txt");

FileWriter fw2 = new FileWriter(file2);

String num;

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

if (Integer.parseInt(num) % 2 == 0) {

fw2.write(num + "\n");

} else {

fw1.write(num + "\n");

}

}

fw1.close();

fw2.close();

} catch (Exception e) {

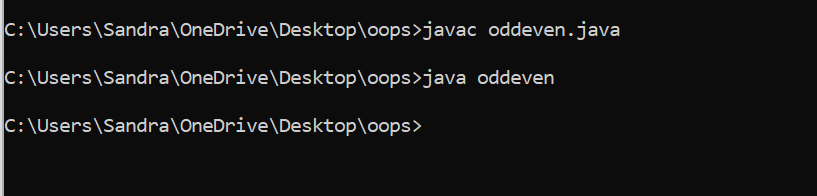
System.out.println("Error");

}

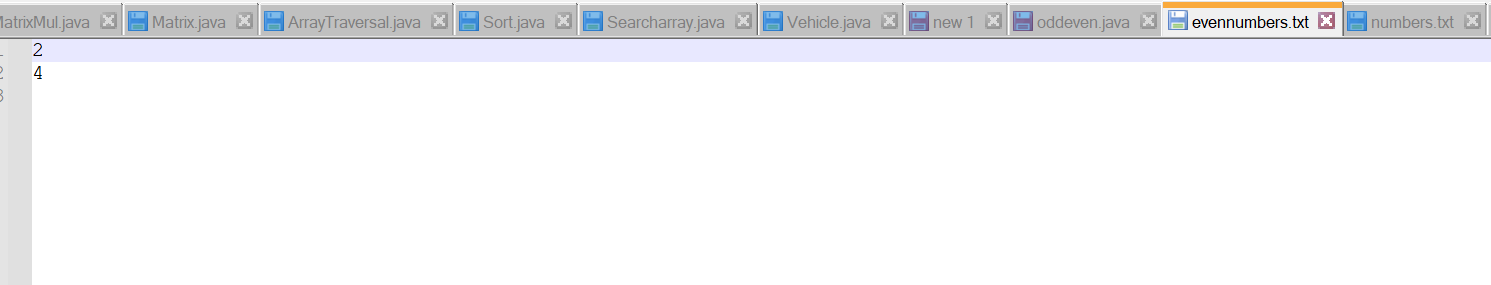
}

}

**output**



evennumbers.txt



oddnumbers.txt

