

**OBJECT ORIENTED**

**PROGRAMMING**

**NAME:AYUSH MENDIRATTA**

**CSE 3B G1**

**2K21CSUN01057**

**SUBMITTED TO:MR.RAM SIR**

**Lab – 1**

1. **Declare an integer variable called n.**

**Ans.** int n;

1. **Set the n to 5;**

n=5;

1. **Declare an integer array called a**

int []a = new int[n];

**4. Write a loop that initializes a to the squares of the first five integers.**

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

int sq = i+1;

a[i] = sq\*sq;

}

for(int b:a) {

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

}

**5. Write a loop that calculates the average of the values of a.**

System.out.println("\nFor Average:- ");

int sum = 0;

for(int b:a) {

sum+=b;

}

System.out.println("Average is "+ sum/a.length);

**6. Write a loop that will move the items in array a into another array called b so that the order of b is the reverse of the order of a.**

int b[] = new int[5];

int num =0;

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

b[num] = a[i];

num=num+1;

}

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

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

}

**7. Write a loop that will calculate the first 10 Fibonacci numbers.**

int fib = 10;

int aa = 0;

int bb = 1;

int cc;

System.out.println(aa+" "+bb);

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

cc = aa+bb;

aa = bb;

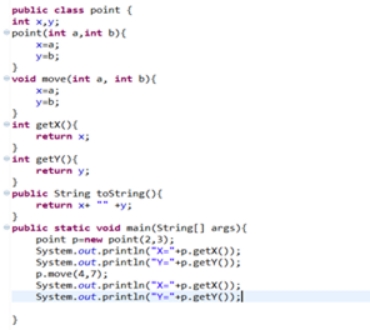
bb = cc;

System.out.print(" " +cc); }

**Lab – 2**

**Q1. Write a java program to design a point class with all the member functions as discussed in the class. Explain the significance of this keyword.**

**Ans. Code:**



**public** **class** point{

**int** x,y;

point(**int** a, **int** b){

x=a;

y=b;

}

**void** move(**int** a, **int** b) {

x=a;

y=b;

}

**int** getX() {

**return** x;

}

**int** getY() {

**return** y;

}

**public** String toString() {

**return** x+""+y;

}

**public** **static** **void** main(String[] args) {

point p= **new** point(2,3);

System.***out***.println("X=",+p.getX());

System.***out***.println("Y="+p.getY());

p.move(4,7);

System.***out***.println("X="+p.getX());

System.***out***.pritln("Y="+p.getY());

}

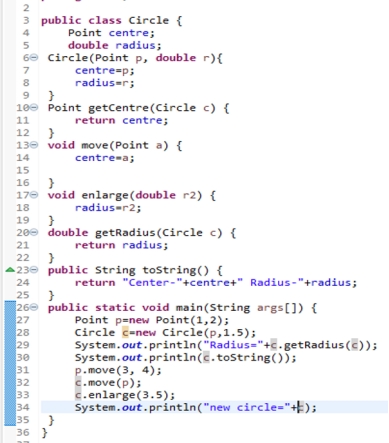
}

**Output:**



**2. Extend the Point Class to create Circle class with necessary members functions**

**Ans. Code:**



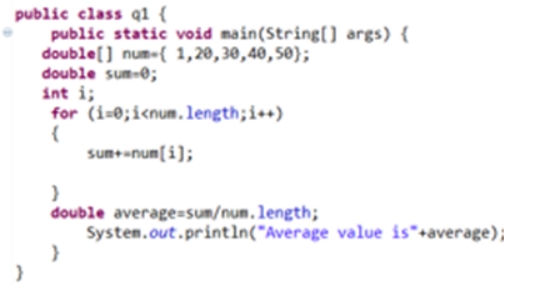
**Output:**



**Lab – 3**

1. **Write a Java program to calculate the average value of array elements.**

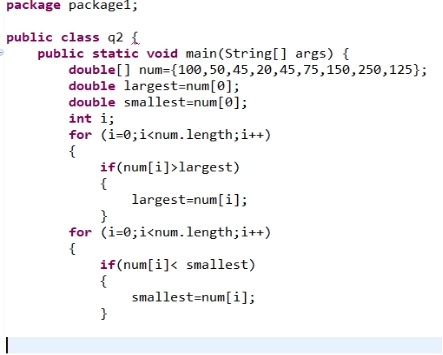
**Ans. Code:**



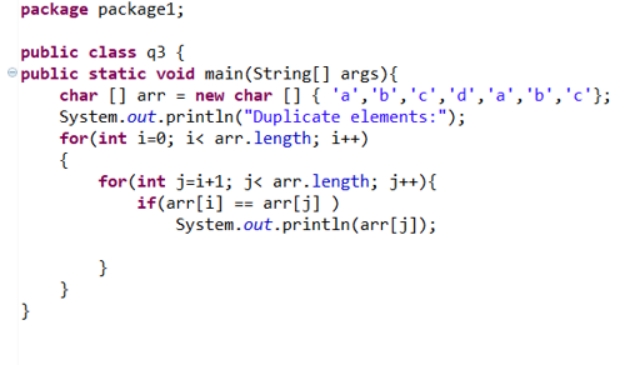
**Output:**

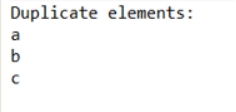


1. **Write a Java program to find the maximum and minimum value of an array.**

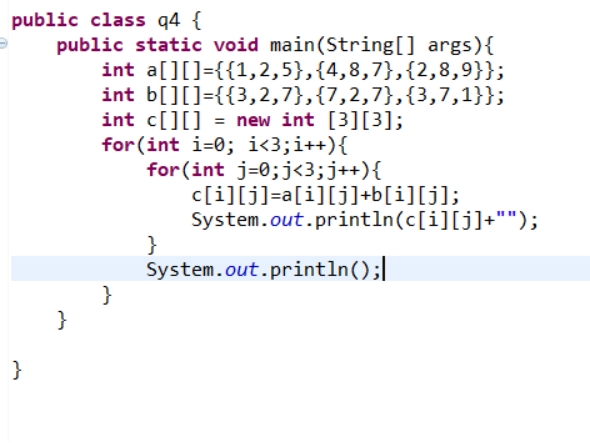


**Write a Java program to find duplicate.**





**4. Write a Java program to add two matrices of the same size**



LAB 4

Question1 :

Code:

public class cml1 {

    public static void main(String[] args)

    {

        double a=Double.parseDouble(args[0]);

        double b=Double.parseDouble(args[1]);

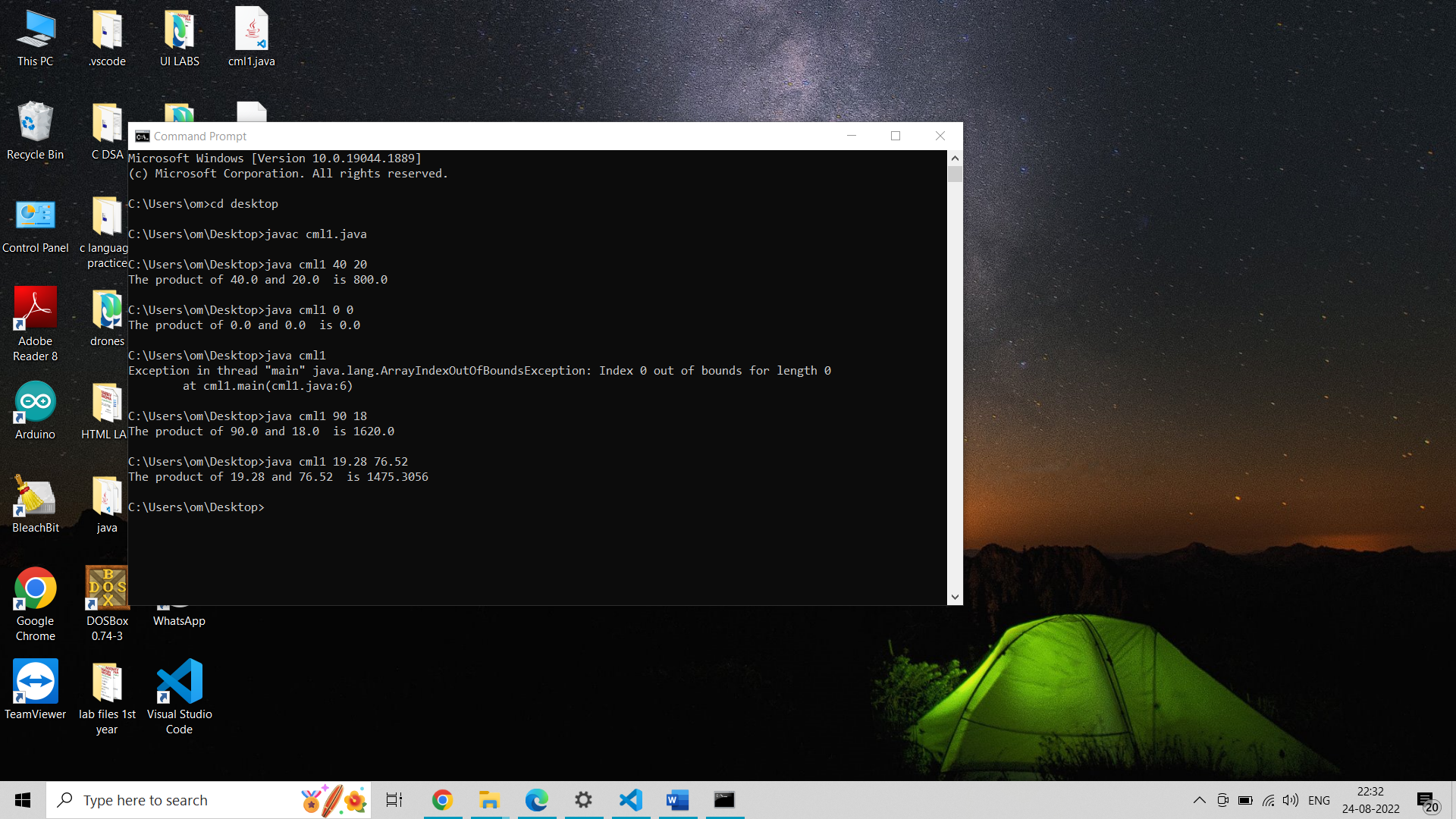
        double result = a\*b;

        System.out.println("The product of "+a+" and "+b+"  is "+result);

    }

}

Output:



Question 2:

Code:

public class cml2 {

    public static void main(String args[]){

    int a =Integer.parseInt(args[0]);

    if (a%2==0){

        System.out.println(" The Number "+a+" is even");

    }

    else{

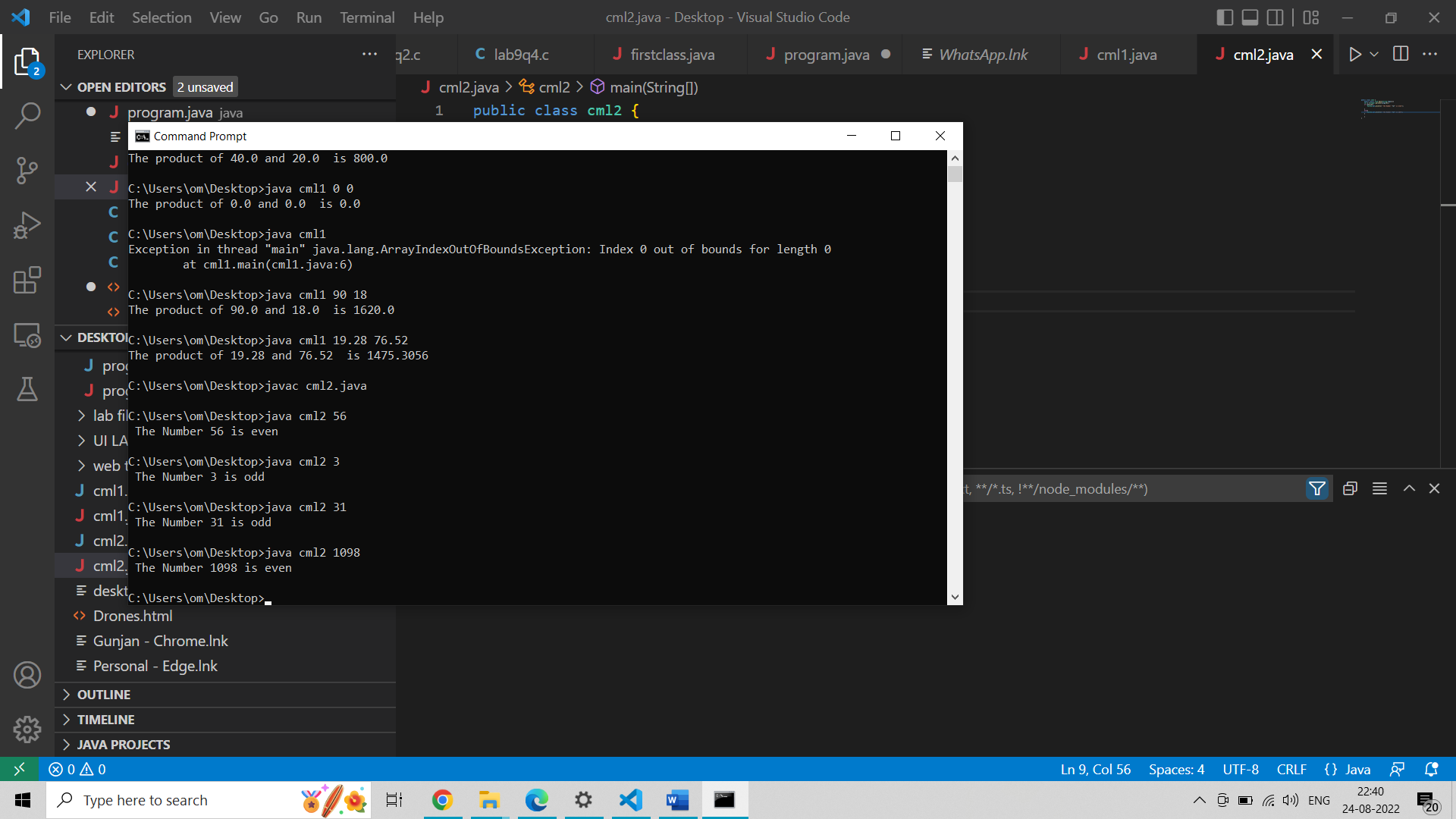
        System.out.println(" The Number "+a+" is odd");

    }

    }

}

Output:



Question3:

Code:

public class cml3 {

    public static void main(String args[]){

        double radius =Double.parseDouble(args[0]);

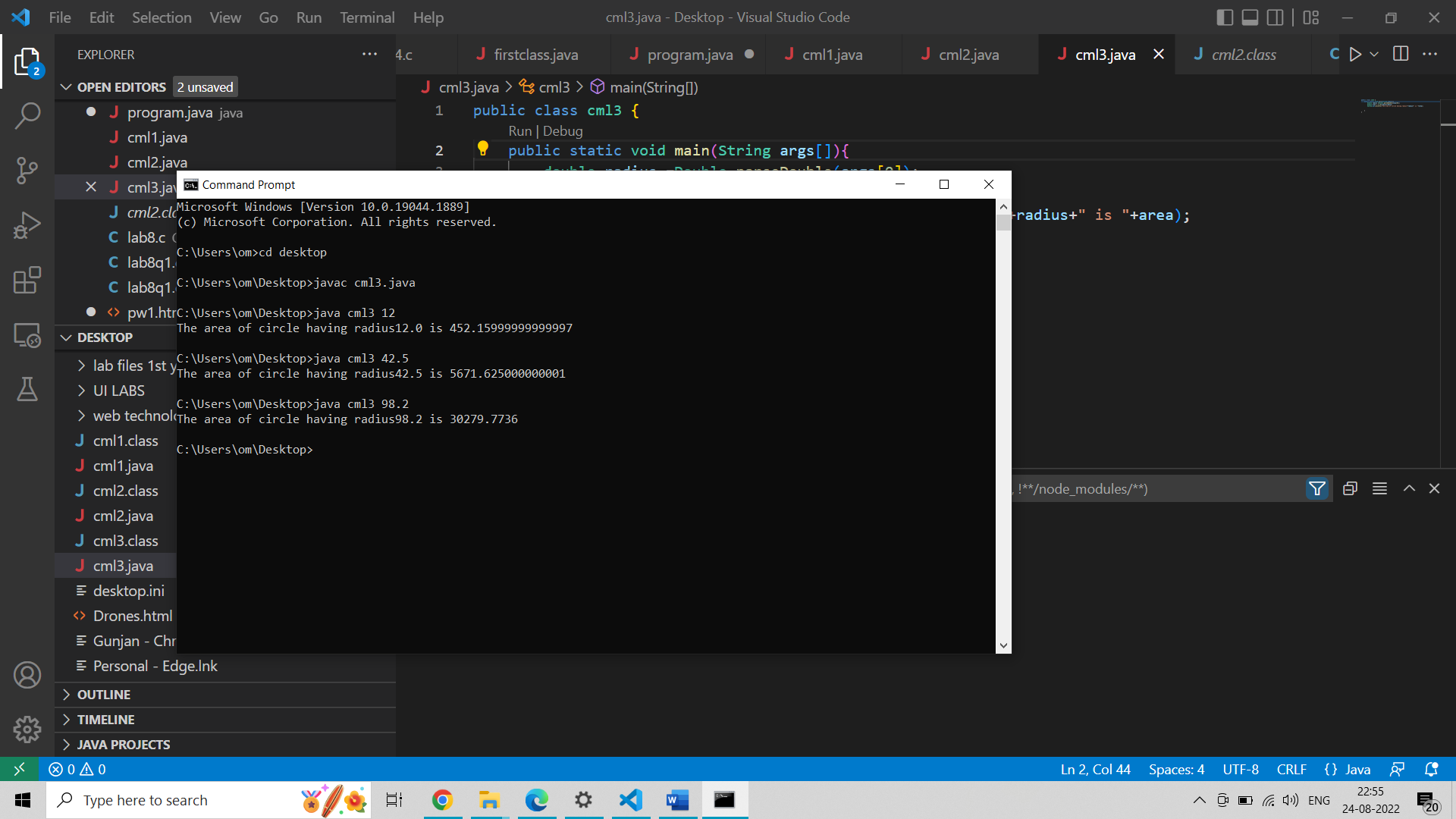
        double area = 3.14\*radius\*radius;

        System.out.println("The area of circle having radius"+radius+" is "+area);

    }

}

Output:



Question 4:

Code:

public class  cml4 {

    public static void main(String args[]){

        int length = args.length;

        for(int i = 0;i<length; i++)

            {

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

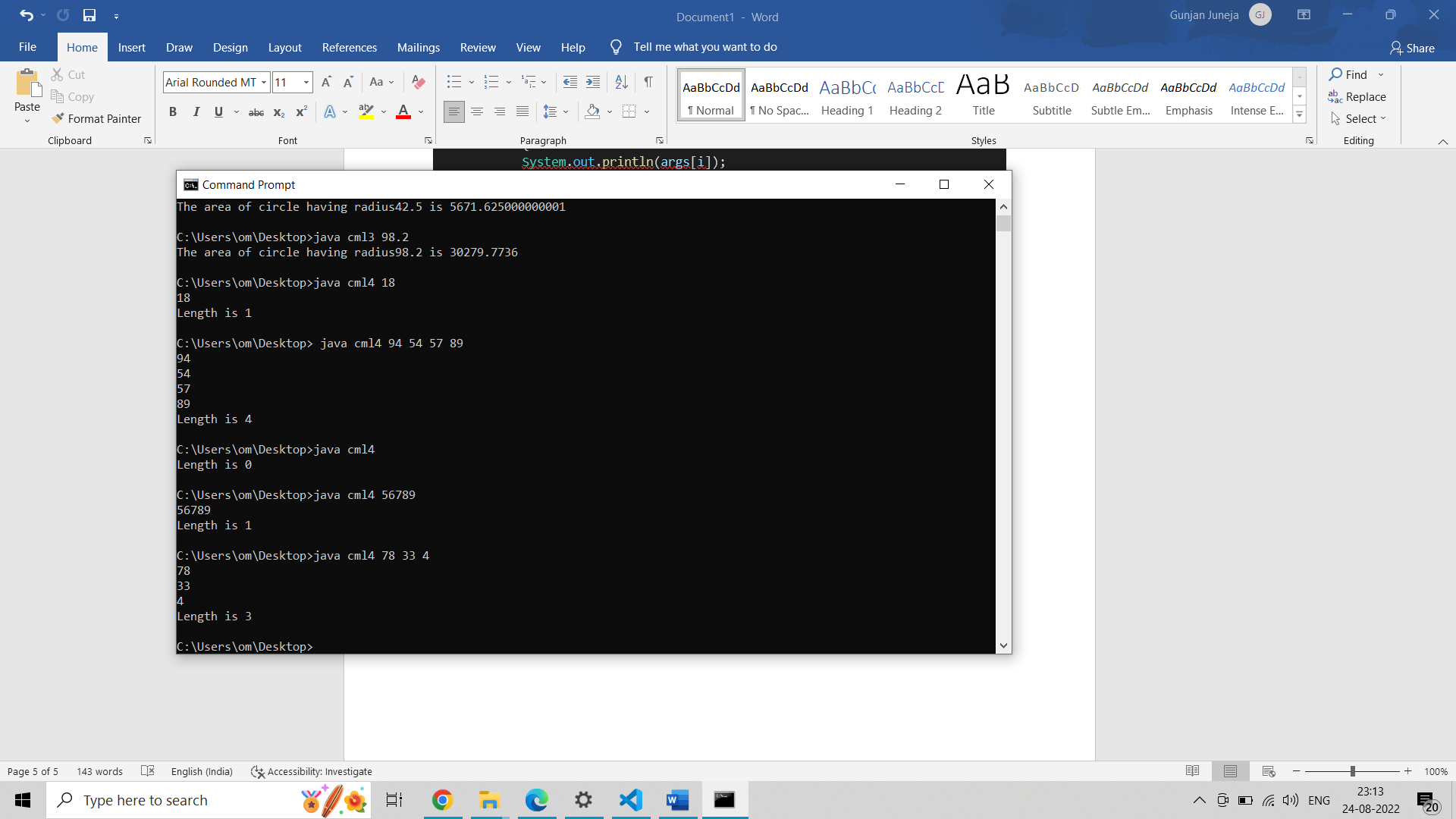
            }

        System.out.println("Length is " + args.length);

    }

}

Output:



Question 5:

Code:

public class person {

    private String name;

    private int age;

    private double salary;

    public void setname(String pname){

         name =pname;

    }

    public void setage(int n){

        age=n;

    }

    public void setsalary(double income){

        salary =income;

    }

    public void print(){

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

System.out.println("Age:" + age);

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

    }

    public static void main(String args[]){

    person person1=new person() ;

    person1.setname("KAJAL");

    person1.setage(19);

    person1.setsalary(0.0);

    person1.print();

    }

}

Question 6:

Code:

class rectangle {

    private double length = 00.00;

    private double breadth = 00.00;

    public double  getarea(double x, double y) {

      return x\*y;

    }

    public static void main(String[] args) {

      rectangle rect = new rectangle();

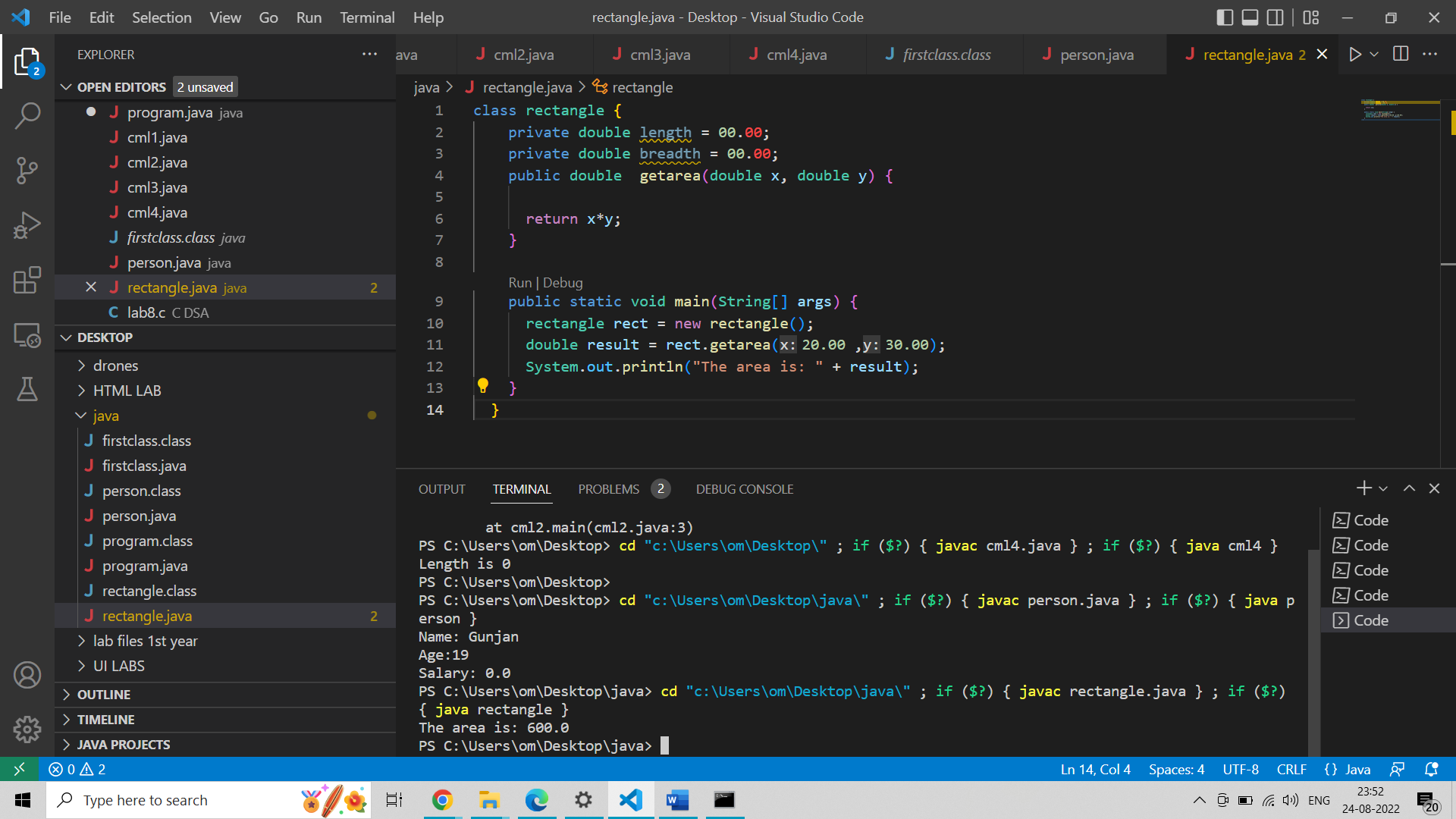
      double result = rect.getarea(20.00 ,30.00);

      System.out.println("The area is: " + result);

    }

  }

Output:



**Lab – 6**

1. **Define a class MyNumber having one private int data member. Write a default constructor to initialize it to 0 and another constructor to initialize it to a value (Use this).**

**Write methods isNegative, isPositive, isZero, isOdd, isEven. Create an object in main. Use Scanner class to pass a value to the object .**

**Ans. Code:**package lab6;

import java.util.Scanner;

class MyNumber{

private int data;

MyNumber(){

this.data = 0;

System.out.println("data = "+ this.data);

}

MyNumber(int value){

this.data = value;

System.out.println("data = "+ this.data);

}

void isnegetive(){

if (this.data < 0){

System.out.println("The value is negetive");

}

}

void ispositive(){

if (this.data > 0){

System.out.println("The value is positive");

}

}

void iseven(){

if (this.data %2 == 0){

System.out.println("The value is even");

}

}

void isodd(){

if (this.data %2 != 0){

System.out.println("The value is odd");

}

Name – Sarthak Pant

}

}

public class Main {

public static void main(String[] args) {

MyNumber obj = new MyNumber();

Scanner s = new Scanner(System.in);

System.out.println("enter the value for the data : ");

int val = s.nextInt();

MyNumber obj2 = new MyNumber(val);

obj2.isnegetive();

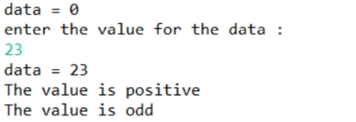
obj2.ispositive();

obj2.iseven();

obj2.isodd();

}

}

**Output:**

1. **Define a Student class (roll number, name, percentage). Define a default and parameterized constructor. Override the toString method. Keep a count objects created. Create objects using parameterized constructor and display the object count after each object is created. (Use static member and method). Also display the contents of each object.**

**Ans. Code:**

package lab6;

import java.util.Scanner;

class Student{

int rollno;

String name = "abc";

double percentage;

static int count = 0;

Student(){

count++;

this.name = "abc";

System.out.println("roll : "+this.rollno);

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

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

}

Student(int roll,String nam, double perc){

count++;

this.rollno = roll;

this.name = nam;

this.percentage = perc;

System.out.println("roll : "+this.rollno);

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

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

}

static void counter(){

System.out.println("total number of objects : "+count);

}

}

public class Main2 {

public static void main(String[] args) {

Student ob1 = new Student();

Student ob2 = new Student(01,"MIKE",97.56);

Student ob3 = new Student(02,"PETER",98.99);

Student ob4 = new Student(03,"SUBARU",79);

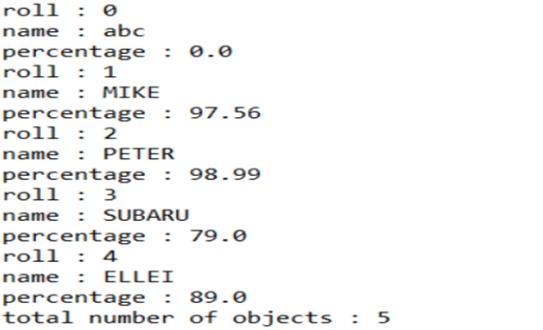
Student ob5 = new Student(04,"ELLEI",89);

Student.counter();

}

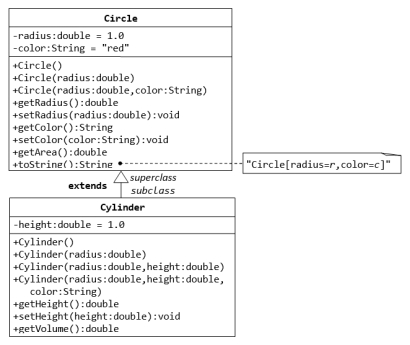
}

**Output:**



**Lab – 7**

**Q1. A class called circle is designed as shown in the following class diagram. It contains:**

* **Two private instance variables: radius (of the type double) and color (of the type String), with default value of 1.0 and "red", respectively.**
* **Three *overloaded* constructors - a *default* constructor with no argument, a constructor which takes a double argument for radius and a constructor which takes a double argument for radius and String argument for Color.**
* **In this exercise, a subclass called Cylinder is derived from the superclass Circle as shown in the class diagram**
* **use super() and super(radius) and inherits the variables and methods from the superclass Circle**

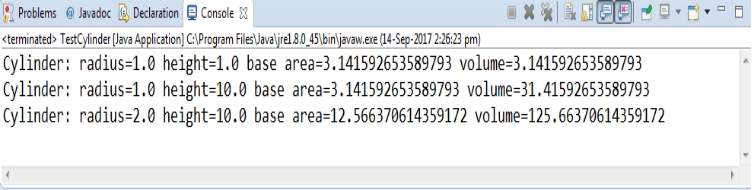
**Write a test program (says TestCylinder) to test the Cylinder class created. Create three objects of Cylinder class namely c1,c2 and c3 as below and display height,radius, area and volume of all the objects**

Cylinder c1 = new Cylinder();

Cylinder c2 = new Cylinder(10.0);

Cylinder c3 = new Cylinder(2.0, 10.0);

**Require Output:**



**Ans. Code:**

**public** **class** Circle {

**private** **double** radius;

**private** String color;

**public** Circle() {

radius = 1.0;

color = "red";

}

**public** Circle(**double** r) {

radius = r;

color = "red";

}

**public** **double** getRadius() {

**return** radius;

}

**public** **double** getArea() {

**return** radius\*radius\*Math.***PI***;

}

}

**public** **class** Cylinder **extends** Circle {

**private** **double** height;

**public** Cylinder() {

**super**();

height = 1.0;

}

**public** Cylinder(**double** height) {

**super**();

**this**.height = height;

}

**public** Cylinder(**double** radius, **double** height) {

**super**(radius);

**this**.height = height;

}

**public** **double** getHeight() {

**return** height;

}

**public** **double** getVolume() {

**return** getArea()\*height;

}

}

**public** **class** Testcylinder {

**public** **static** **void** main (String[] args) {

Cylinder c1 = **new** Cylinder();

System.***out***.println("Cylinder:"

+ " radius=" + c1.getRadius()

+ " height=" + c1.getHeight()

+ " base area=" + c1.getArea()

+ " volume=" + c1.getVolume());

Cylinder c2 = **new** Cylinder(2.0, 10.0);

System.***out***.println("Cylinder:"

+ " radius=" + c2.getRadius()

+ " height=" + c2.getHeight()

+ " base area=" + c2.getArea()

+ " volume=" + c2.getVolume());

}

}

**Q2. Code:**

public class person {

String name,address;

person( String name, String address)

{

this.name=name;

this.address=address;

}

String getName()

{

return name;

}

String getAddress() {

return address;

}

void setAddress(String address) {

this.address=address;

}

public String toString()

{

return" name : " +name +" address : "+ address;

}

}

public class staff extends person {

String school ;

double pay;

staff ( String name, String address,String school,double pay)

{

super(name,address);

this.school=school;

this.pay=pay;

}

void setSchool(String school)

{

this.school=school;

}

void setPay(double pay)

{

this.pay=pay;

}

String getSchool()

{

return school;

}

double getPay()

{

return pay;

}

public String toString()

{

return" name : " +name +" address : "+ address + " school : " + school + " pay : " + pay;

}

public static void main (String[] args)

{

staff S1 = new staff( "parkhi" , "faridabad", "mru",200001);

System.out.println(S1.toString());

}

}

public class Student extends person{

String program;

int year;

double fee;

Student ( String name, String address,String program,int year,double fee)

{

super(name,address);

this.program=program;

this.year=year;

this.fee=fee;

}

void setProgram(String program)

{

this.program=program;

}

void setYear(int year)

{

this.year=year;

}

void setFee(double fee)

{

this.fee=fee;

}

String getProgram()

{

return program;

}

int getYear()

{

return year;

}

double getFee()

{

return fee;

}

public String toString()

{

return" name : " +name +" address : "+ address + " program : " + program + " year : " + year + " fee : " + fee;

}

public static void main (String[] args)

{

Student S1 = new Student( "ram" , "faridabad", "b.tech",2020,151000);

System.out.println(S1.toString());

}

}

**Lab – 8**

**1.** **Write a program to implement the following class hierarchy.**

**Code:**

abstract class shape {

private String color;

public shape(String color) {

this.color=color;

}

public String toString() {

return "color="+color;

}

abstract double getArea();

}

class Rectangle extends shape {

private int length;

private int width;

public Rectangle(String color, int length, int width) {

super(color);

this.length=length;

this.width=width;

}

public double getArea() {

return length\*width;

}

public String toString() {

return "Rectangle[length="+length +",width"+width+","+

super.toString()+"]";

}

}

class Triangle extends shape {

private int base;

private int height;

public Triangle(String color,int base,int height) {

super(color);

this.base=base;

this.height=height;

}

public double getArea() {

return (base\*height)/2;

}

public String toString() {

return "Triangle[base="+base+"height="+height+","+super.toString()+"]";

}

}

class TestShape {

public static void main(String[] args) {

shape s1 = new Rectangle("red", 4, 5);

System.out.println(s1);

System.out.println("Area is " + s1.getArea());

shape s2 = new Triangle("blue", 4, 5);

System.out.println(s2);

System.out.println("Area is " + s2.getArea());

}

}

**Q2. Code:**

public interface shape {

abstract double getArea();

}

class Rectangle implements shape {

private int length;

private int width;

public Rectangle(int length, int width) {

this.length = length;

this.width = width;

}

public String toString() {

return "Rectangle[length=" + length + ",width=" + width + "]";

}

public double getArea() {

return length \* width;

}

}

class Triangle implements shape {

private int base;

private int height;

public Triangle(int base, int height) {

this.base = base;

this.height = height;

}

public String toString() {

return "Triangle[base=" + base + ",height=" + height + "]";

}

public double getArea() {

return 0.5 \* base \* height;

}

}

class TestShape {

public static void main(String[] args) {

shape s1 = new Rectangle(1, 2); // upcast

System.out.println(s1);

System.out.println("Area is " + s1.getArea());

shape s2 = new Triangle(3, 4); // upcast

System.out.println(s2);

System.out.println("Area is " + s2.getArea());

}

}

**Lab – 9**

1. **Create a class Student with attributes roll no, name,age and course. Initilize values through parameterized constructor. If age of the student is not in between 15 and 21 then generate user-defined exception “AgeNotWithinRangeException**

public class AgeNotWithinRangeException extends Exception {

public AgeNotWithinRangeException(String msg) {

super(msg);

}

}

public class Student {

private String name;

private int rollno;

private int age;

private String course;

public Student(String name, int rollno, int age, String course) {

this.name = name;

this.rollno = rollno;

this.course = course;

try {

if (age < 15 || age > 21) {

throw new AgeNotWithinRangeException("Age not between 15 and 21");

}

this.age = age;

} catch (AgeNotWithinRangeException e) {

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

}

}

public void display() {

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

System.out.println("Age: " + this.age);

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

System.out.println("Course: " + this.course);

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

}

}

public class TestStudent {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String name = sc.nextLine();

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

int age = Integer.parseInt(sc.nextLine());

System.out.print("Enter roll number: ");

int rollNo = Integer.parseInt(sc.nextLine());

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

String course = sc.nextLine();

Student s1 = new Student(name, rollNo, age, course);

s1.display();

sc.close();

}

}

1. **Write a program which accept two integers and an arithmetic operator from the command line and performs the operation. Fire the following user defined exceptions: i. If the no of arguments are less than 3 then fire “illegalNumberOfArguments”• ii. If the operator is not an Arithmetic operator, throw “InvalidOperatorException” iii. If result is -ve, then throw “NegativeResultException”**

public class IllegalNumberOfAssignmentsException extends Exception {

public IllegalNumberOfAssignmentsException(String msg) {

super(msg);

}

}

public class InvalidOperatorException extends Exception {

public InvalidOperatorException(String msg) {

super(msg);

}

}

public class NegativeResultException extends Exception {

public NegativeResultException(String msg) {

super(msg);

}

}

public class Calc {

public static void main(String[] args) {

try {

if (args.length < 3) {

throw new IllegalNumberOfAssignmentsException("Not enough arguments provided.");

}

int a, b, res = 0;

boolean invalidOperator = false;

a = Integer.parseInt(args[1]);

b = Integer.parseInt(args[2]);

switch (args[0]) {

case "+":

res = a + b;

break;

case "-":

res = a - b;

break;

case "\*":

res = a \* b;

break;

case "/":

res = a / b;

break;

default:

invalidOperator = true;

break;

}

if (invalidOperator)

throw new InvalidOperatorException(args[0] + " is not an operator");

if (res < 0)

throw new NegativeResultException(res + " is negative.");

System.out.printf("%d %s %d = %d\n", a, args[0], b, res);

} catch (Exception e) {

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

}

}

}

1. **Define a class MyDate with members day, month, year. Define default and parameterized constructors. Accept values from the command line and create a date object. Throw user defined – “InvalidDayException” or InvalidMonthException” if the day and month are invalid. If the date is valid display message “valid date”**

