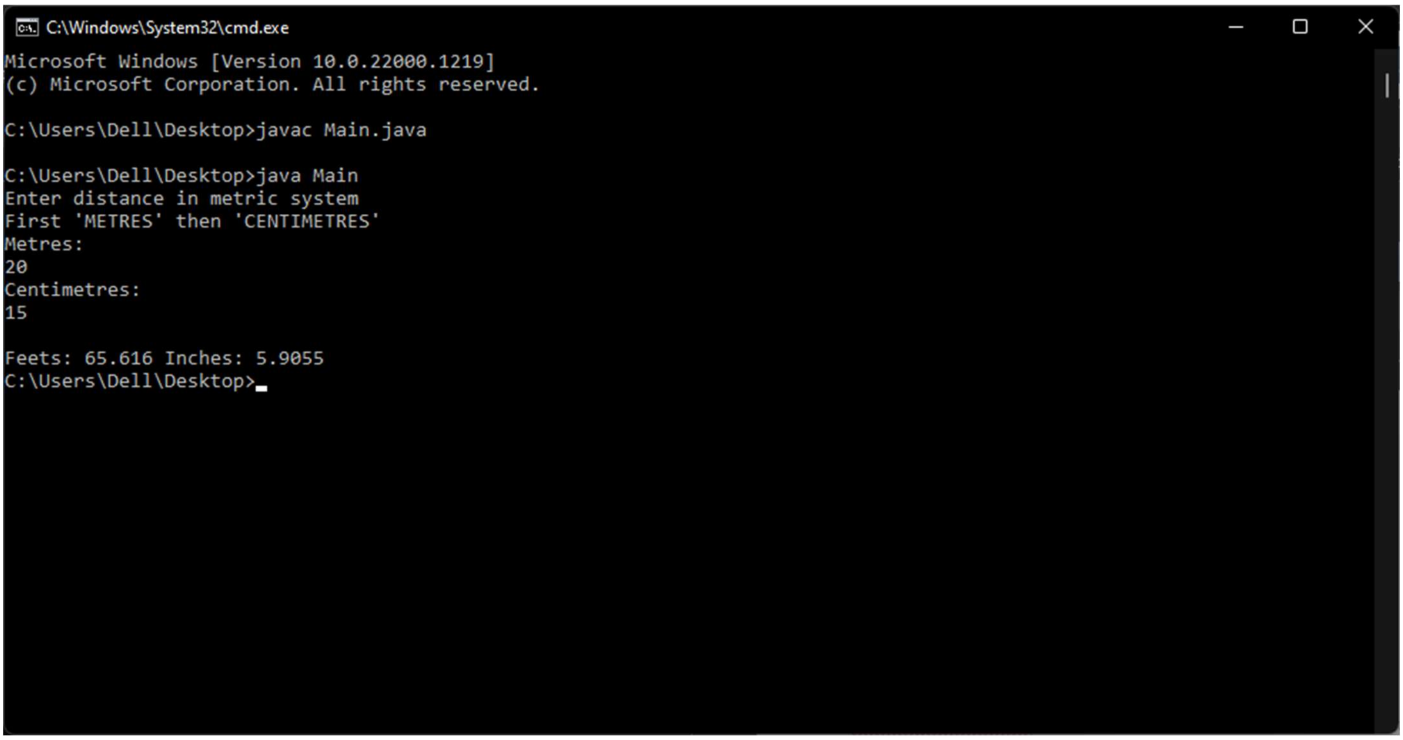
OBJECT ORIENTED PROGRAMMING IN JAVA – MODULE 2

MODULE 2 – Q1

CODE

```java
import java.util.*;
class Distance{
    double feet;
    double inch;
    public void calculateFeetAndInch(int a, int b){
        feet = a*3.2808;
        inch = b*0.3937;
    }
    public void displayFeetAndInch(){
        System.out.print("\nFeets: "+feet+" Inches: "+inch);
    }
}
public class Main{
    public static void main(String[] args) {
        Scanner sc= new Scanner(System.in);
        System.out.print("Enter distance in metric system\nFirst 'METRES' then
'CENTIMETRES'\n");
        System.out.println("Metres:");
        int metres = sc.nextInt();
        System.out.println("Centimetres:");
        int cmetres = sc.nextInt();
        Distance d1 = new Distance();
        d1.calculateFeetAndInch(metres,cmetres);
        d1.displayFeetAndInch();
    }
}
```

OUTPUT



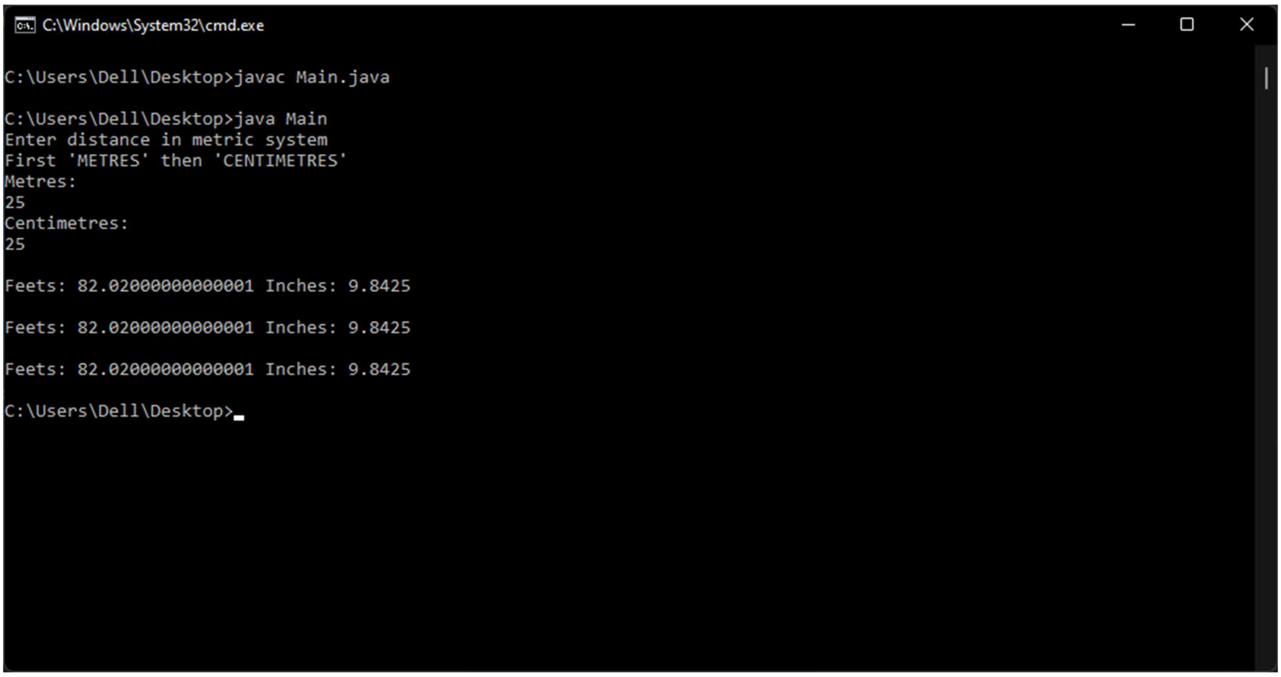
```
Microsoft Windows [Version 10.0.22000.1219]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Enter distance in metric system
First 'METRES' then 'CENTIMETRES'
Metres:
20
Centimetres:
15

Feets: 65.616 Inches: 5.9055
C:\Users\Dell\Desktop>_
```

MODULE 2 – Q2

CODE

```
import java.util.*;
class Distance2 implements Cloneable{
    double feet, inch;
    Distance2(int a, int b){
        feet = a*3.2808;
        inch = b*0.3937;
    }
    public void display(){
        System.out.println("\nFeets: "+feet+" Inches: "+inch);
    }
    public Object clone() throws CloneNotSupportedException{
        return super.clone();
    }
}
public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter distance in metric system\nFirst 'METRES' then
'CENTIMETRES'\n");
        System.out.println("Metres:");
        int metres = sc.nextInt();
        System.out.println("Centimetres:");
        int cmetres = sc.nextInt();
        //creating an object-1
        Distance2 d1 = new Distance2(metres,cmetres);
        d1.display();
        //creating object-2 as a reference variable
        Distance2 d2 = d1;
        d2.display();
        try{
            Distance2 d3 = (Distance2)d1.clone();
            d3.display();
        }
        catch (Exception e){
            System.out.println(e);
        }
    }
}
```

OUTPUT



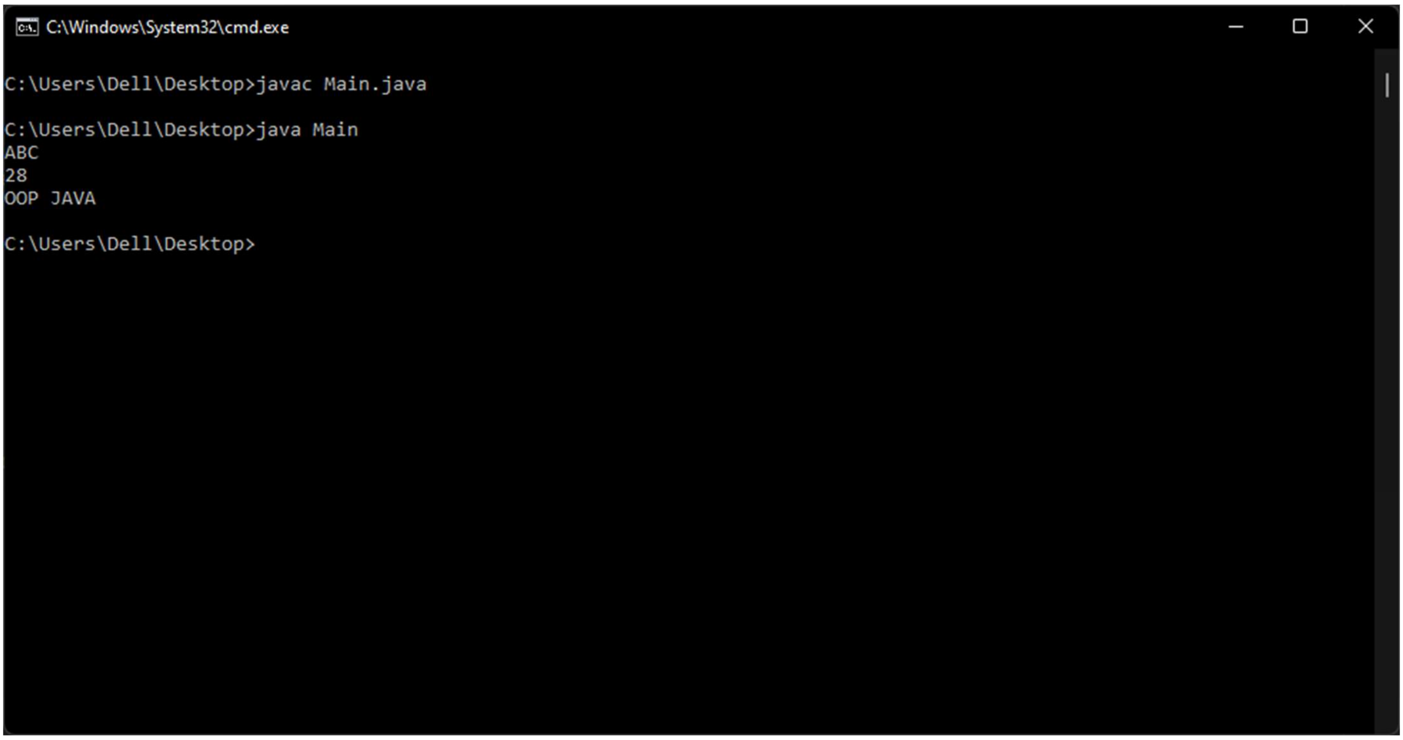
```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Enter distance in metric system
First 'METRES' then 'CENTIMETRES'
Metres:
25
Centimetres:
25

Feets: 82.02000000000001 Inches: 9.8425

Feets: 82.02000000000001 Inches: 9.8425

Feets: 82.02000000000001 Inches: 9.8425

C:\Users\Dell\Desktop>_
```

MODULE 2 – Q3

CODE

```
class Employee{
    String name;
    Employee(){
        name="ABC";
    }
}
class Teacher extends Employee{
    int age;
    String subject;
    Teacher(){
        super();
        age=28;
        subject="OOP JAVA";
    }
}
public class Main{
    public static void main(String[] args) {
        Teacher t1 = new Teacher();
        System.out.println(t1.name);
        System.out.println(t1.age);
        System.out.println(t1.subject);
    }
}
```

OUTPUT



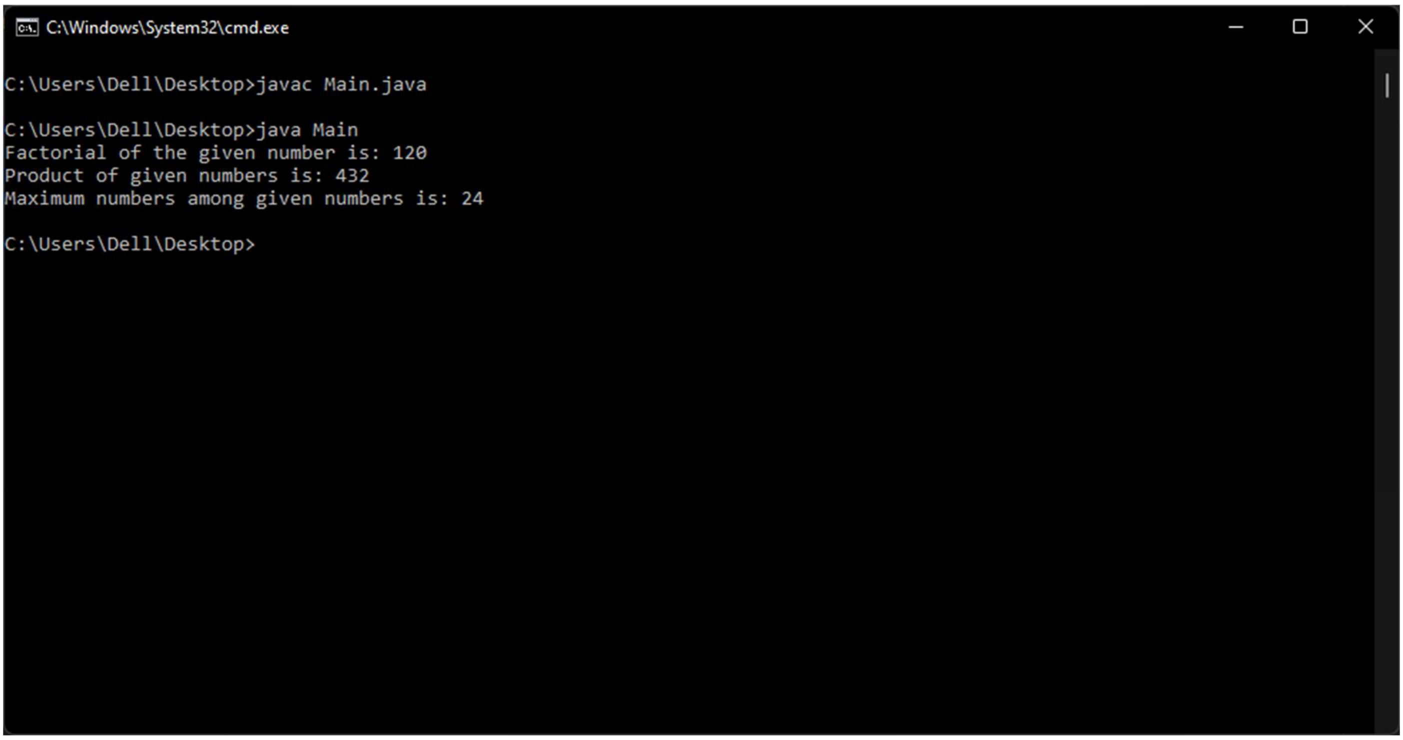
```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
ABC
28
OOP JAVA

C:\Users\Dell\Desktop>
```

MODULE 2 – Q4

CODE

```
import java.util.*;
class MultiConstructor{
    int ans;
    int max;
    MultiConstructor(int a){
        int fact=1;
        for (int i=1;i<=a;i++){
            fact=fact*i;
        }
        ans=fact;
    }
    MultiConstructor(int a, int b){
        ans=a*b;
        if (a>b){
            max=a;
        }
        else{
            max=b;
        }
    }
    MultiConstructor(MultiConstructor m){ //copy constructor
        ans=m.ans;
        max=m.max;
    }
}
public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        MultiConstructor m1=new MultiConstructor(5);
        System.out.println("Factorial of the given number is: "+m1.ans);
        MultiConstructor m2=new MultiConstructor(18,24);
        System.out.println("Product of given numbers is: "+m2.ans);
        MultiConstructor m3=new MultiConstructor(m2);
        System.out.println("Maximum numbers among given numbers is: "+m3.max);
    }
}
```

OUTPUT



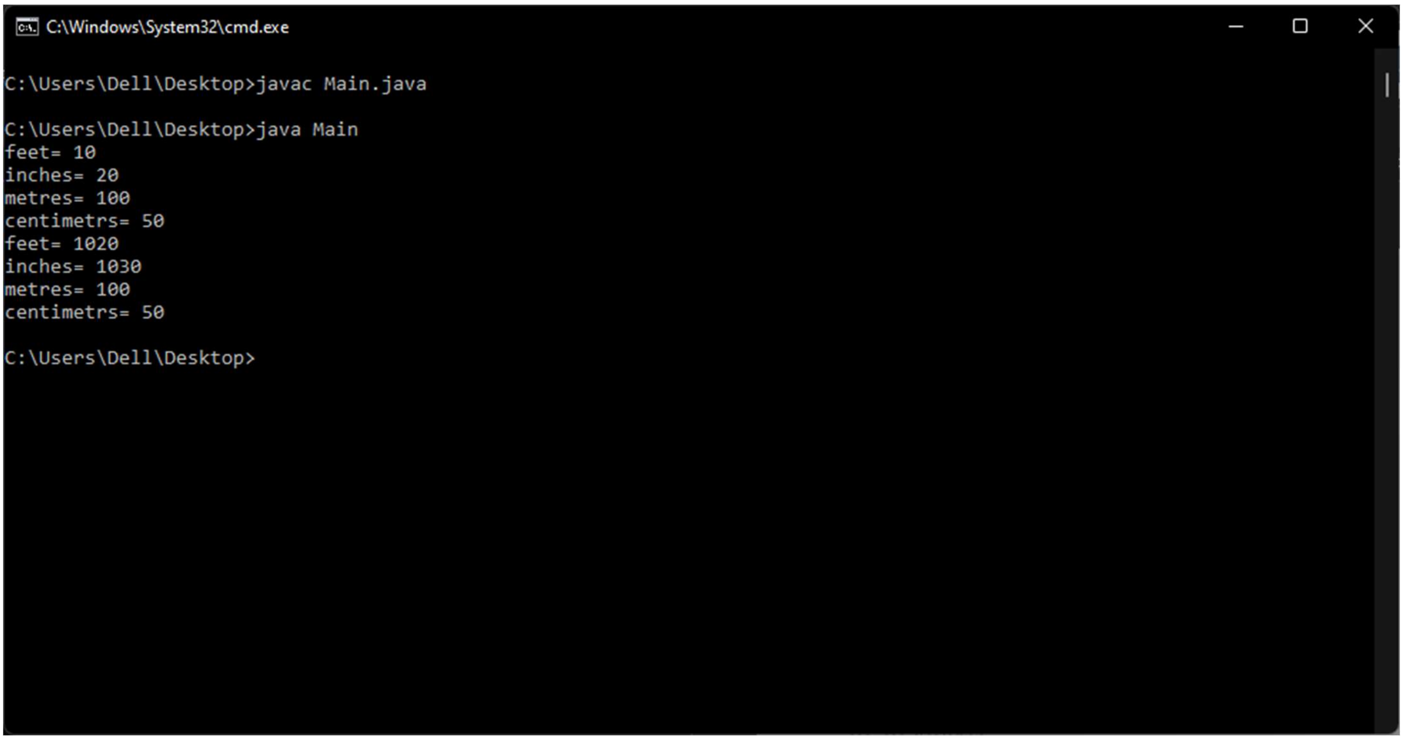
```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Factorial of the given number is: 120
Product of given numbers is: 432
Maximum numbers among given numbers is: 24

C:\Users\Dell\Desktop>
```

MODULE 2 – Q5

CODE

```
class Distance {
    public int feet, inches;
    private int metrs, centimetrs;
    public final int kilo = 1000;//final keyword
    Distance(int f, int i) {
        metrs = 100;
        centimetrs = 50;
        this.feet = f;
        this.inches = i;
    }
    void display() {
        System.out.println("feet= " + feet);
        System.out.println("inches= " + inches);
        System.out.println("metres= " + metrs);
        System.out.println("centimetrs= " + centimetrs);
    }
    void addDistance(Distance d3) {
        // kilo=2000; //because of final value can't be assigned
        d3.feet = d3.feet + kilo;
        d3.inches = d3.inches + kilo;
    }
}
public class Main {
    public static void main(String args[]) {
        Distance d1 = new Distance(10, 20);
        Distance d3 = new Distance(20, 30);
        d1.display();
        // System.out.println("metres= "+d1.metrs);
        // metrs can't be accessed because of private access specifier
        // System.out.println("centimetres= "+d1.centimetrs);
        // centimeters can't be accessed because of private access specifier
        d1.addDistance(d3);
        d3.display();
    }
}
```

OUTPUT



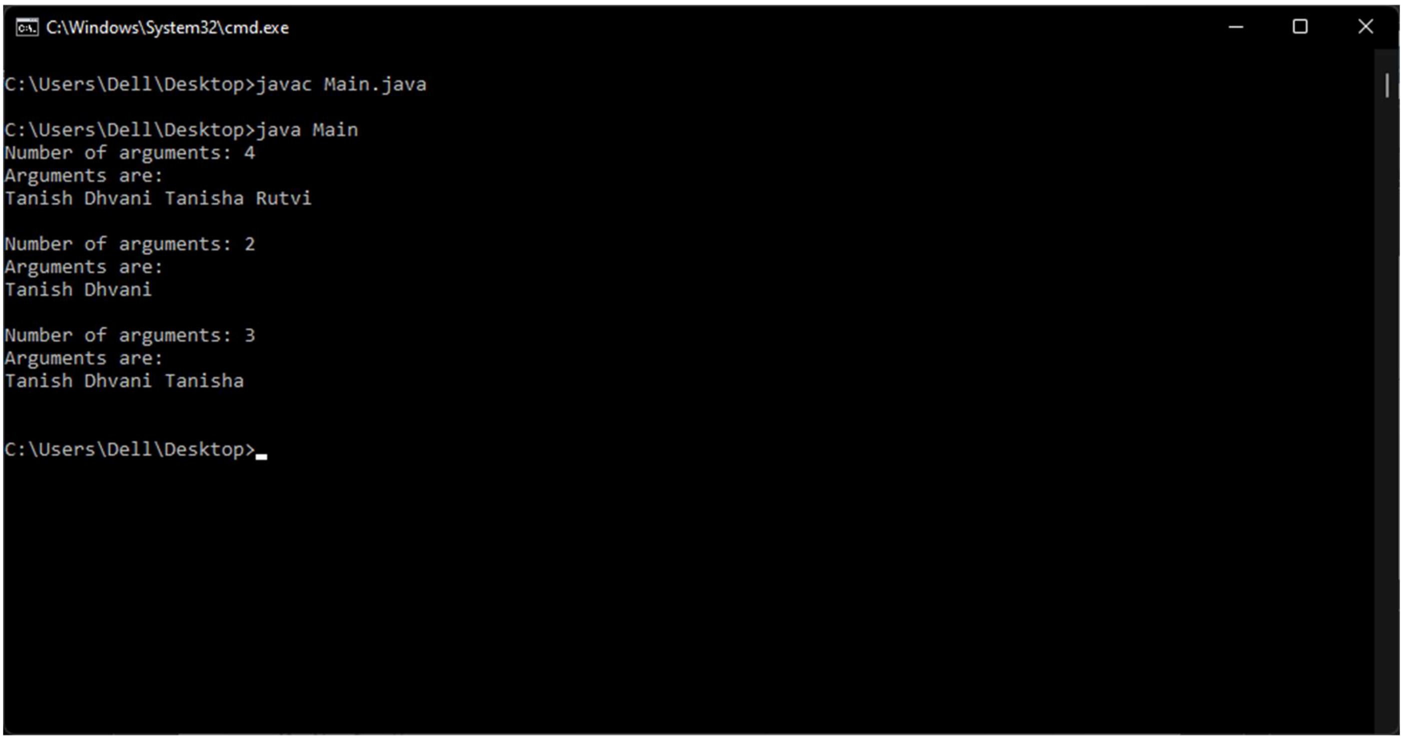
```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
feet= 10
inches= 20
metres= 100
centimetrs= 50
feet= 1020
inches= 1030
metres= 100
centimetrs= 50

C:\Users\Dell\Desktop>
```

MODULE 2 – Q6

CODE

```
import java.util.*;
public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Varargs("Tanish","Dhvani","Tanisha","Rutvi");
        Varargs("Tanish","Dhvani");
        Varargs("Tanish","Dhvani","Tanisha");
    }
    public static void Varargs(String...s1){
        int a=s1.length;
        System.out.println("Number of arguments: "+a);
        System.out.println("Arguments are:");
        for (String i:s1){
            System.out.print(i+" ");
        }
        System.out.print("\n\n");
    }
}
```

OUTPUT



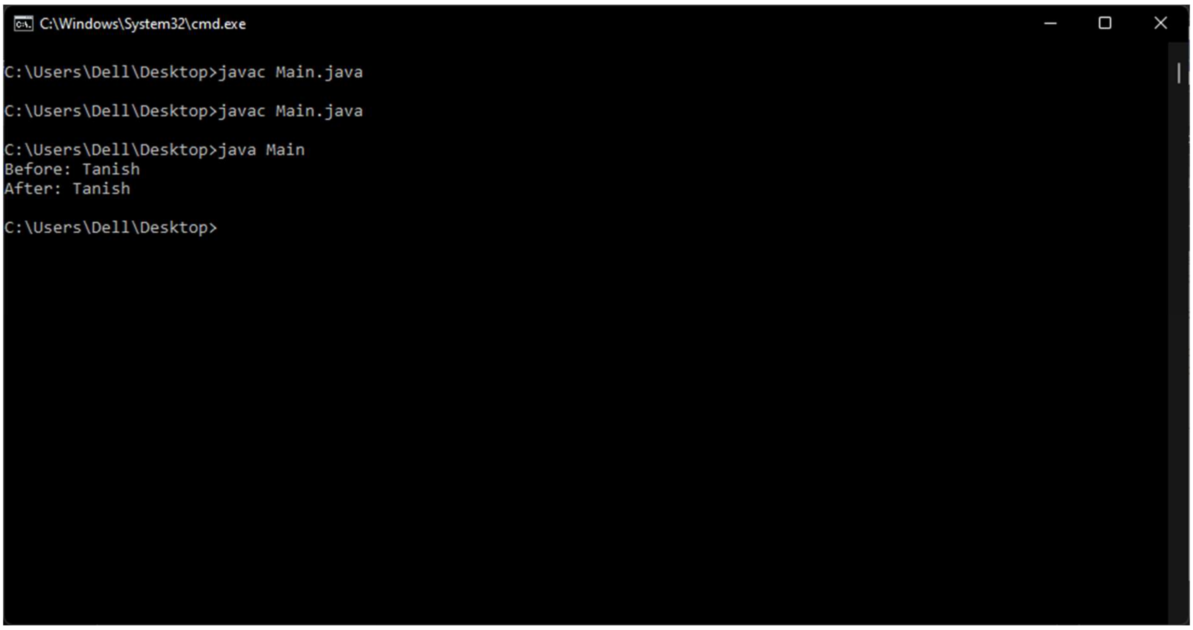
```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Number of arguments: 4
Arguments are:
Tanish Dhvani Tanisha Rutvi

Number of arguments: 2
Arguments are:
Tanish Dhvani

Number of arguments: 3
Arguments are:
Tanish Dhvani Tanisha


C:\Users\Dell\Desktop>
```

MODULE 2 – Q7

CODE

```
//pass by value
import java.util.*;
class Pass_By_Value{
    String s;
}
public class Main{
    static void changeName(String st){
        st="Now this is the new name!";
    }
    public static void main(String[] args) {
        Pass_By_Value obj = new Pass_By_Value();
        obj.s="Tanish";
        System.out.println("Before: "+obj.s);
        changeName(obj.s);
        System.out.println("After: "+obj.s);
    }
}
```

OUTPUT

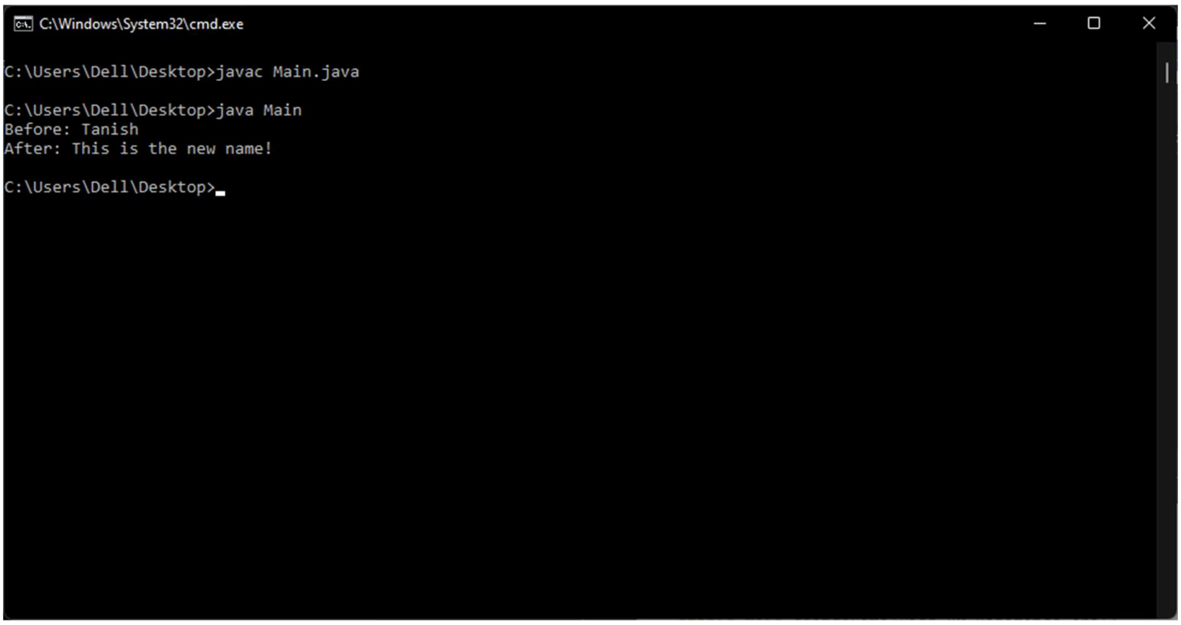


```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Before: Tanish
After: Tanish

C:\Users\Dell\Desktop>
```

CODE

```
//pass by reference
import java.util.*;
class Pass_By_Reference{
    String s;
}
public class Main{
    static void changeName(Pass_By_Reference obj){
        obj.s="This is the new name!";
    }
    public static void main(String[] args) {
        Pass_By_Reference obj = new Pass_By_Reference();
        obj.s="Tanish";
        System.out.println("Before: "+obj.s);
        changeName(obj);
        System.out.println("After: "+obj.s);
    }
}
```

OUTPUT

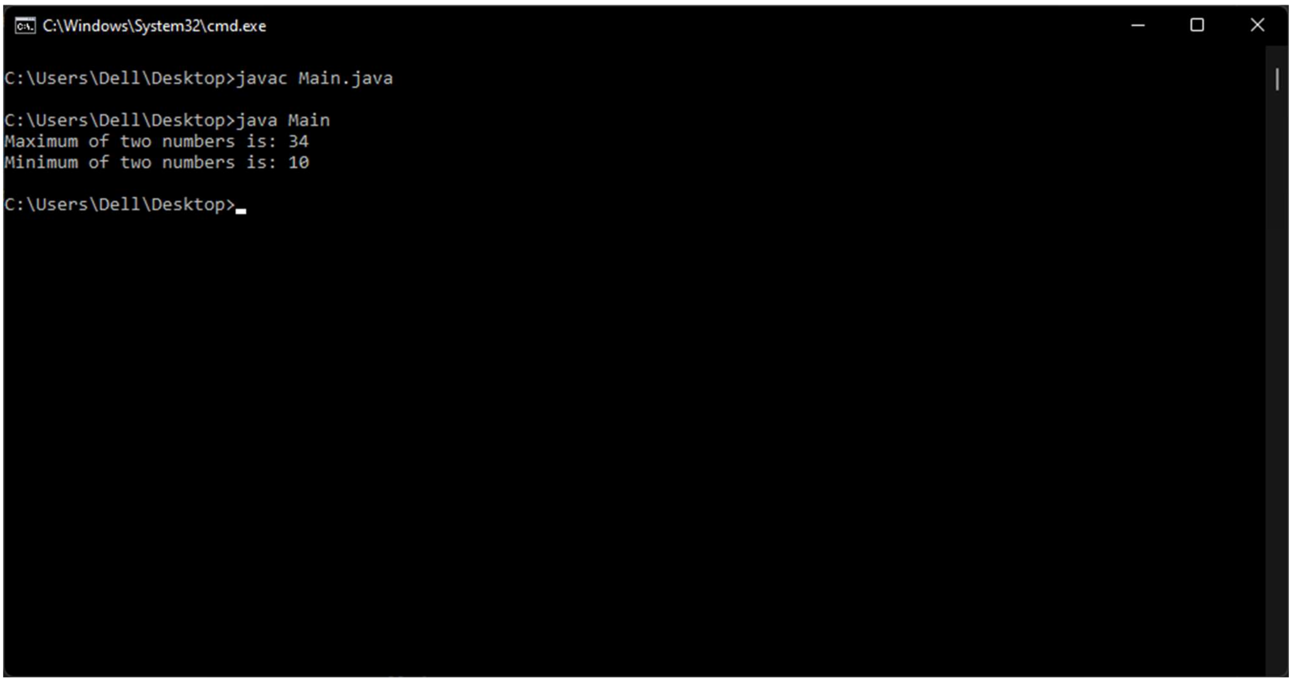


```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Before: Tanish
After: This is the new name!

C:\Users\Dell\Desktop>
```

CODE

```
import java.util.*;
class Return_values{
    int a;
    Return_values(int a1){
        a=a1;
    }
}
public class Main{
    static int findMax(Return_values o1, Return_values o2){
        if (o1.a>o2.a){
            return o1.a;
        }
        else{
            return o2.a;
        }
    }
    static int findMin(Return_values o1, Return_values o2){
        if (o1.a<o2.a){
            return o1.a;
        }
        else{
            return o2.a;
        }
    }
    public static void main(String args[]){
        Return_values obj1= new Return_values(10);
        Return_values obj2= new Return_values(34);
        System.out.println("Maximum of two numbers is: " + findMax(obj1,obj2));
        System.out.println("Minimum of two numbers is: " + findMin(obj1,obj2));
    }
}
```

OUTPUT



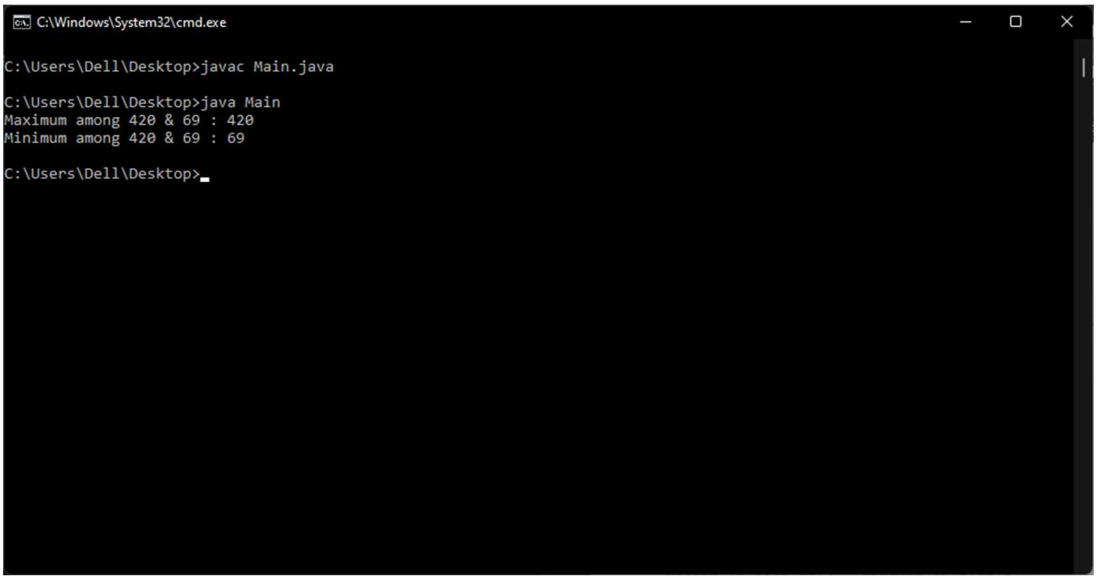
```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Maximum of two numbers is: 34
Minimum of two numbers is: 10

C:\Users\Dell\Desktop>_
```

CODE

```
import java.util.*;
class Return_Objects{
    int a;
    Return_Objects(int a1){
        a=a1;
    }
}
```

```
public class Main{
    static Return_Objects findMax(Return_Objects o1, Return_Objects o2){
        int max;
        if (o1.a > o2.a){
            max=o1.a;
        }
        else{
            max=o2.a;
        }

        Return_Objects temp = new Return_Objects(max);
        return temp;
    }
    static Return_Objects findMin(Return_Objects o1, Return_Objects o2){
        int min;
        if (o1.a < o2.a){
            min=o1.a;
        }
        else{
            min=o2.a;
        }
        Return_Objects temp = new Return_Objects(min);
        return temp;
    }
    public static void main(String[] args) {
        Return_Objects obj1 = new Return_Objects(420);
        Return_Objects obj2 = new Return_Objects(69);
        Return_Objects obj3 = findMax(obj1,obj2);
        Return_Objects obj4 = findMin(obj1,obj2);
        System.out.println("Maximum among "+obj1.a+" & "+obj2.a+" : "+obj3.a);
        System.out.println("Minimum among "+obj1.a+" & "+obj2.a+" : "+obj4.a);
    }
}
```

OUTPUT



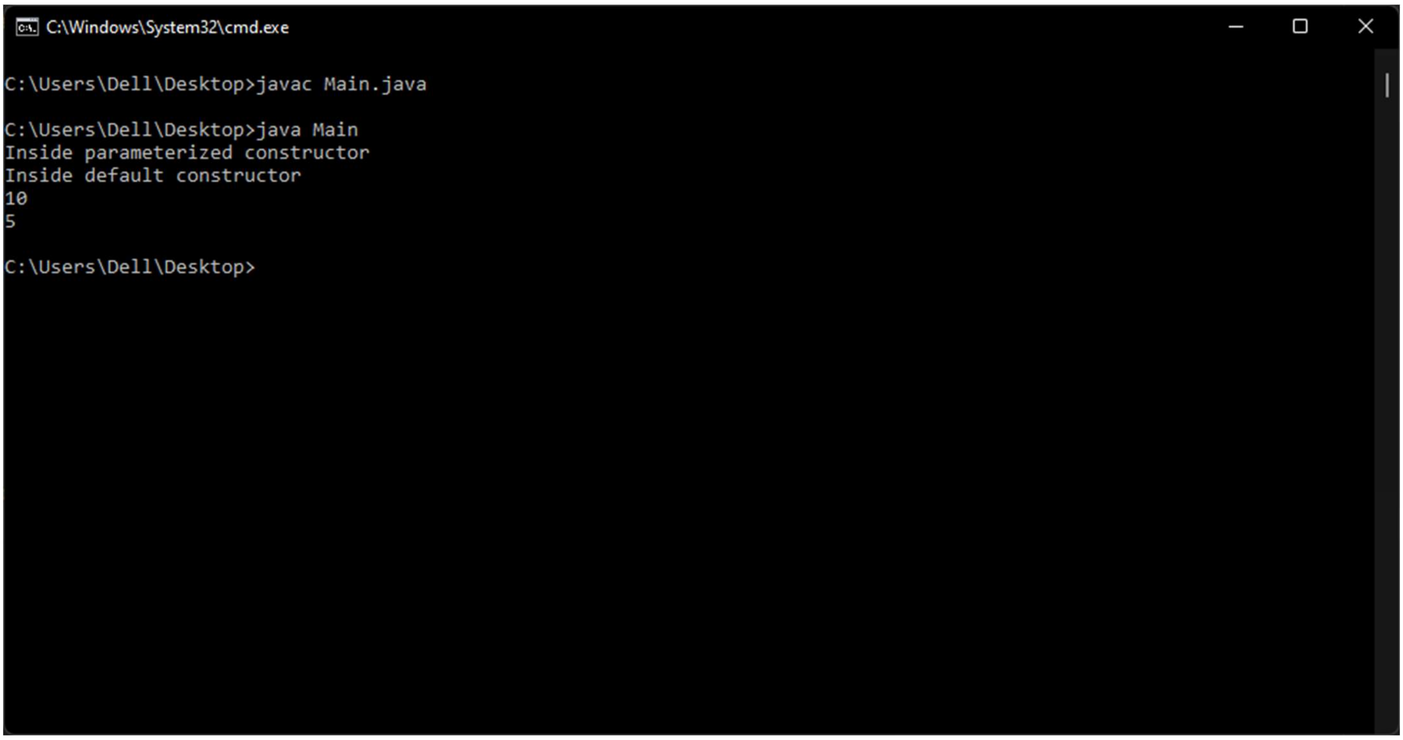
```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Maximum among 420 & 69 : 420
Minimum among 420 & 69 : 69

C:\Users\Dell\Desktop>
```

MODULE 2 – Q8

CODE

```
import java.util.*;
class ConstructorClass{
    int a;
    int b;
    //default constructor
    ConstructorClass(){
        this(10,5);
        System.out.println("Inside default constructor");
    }
    //parameterized constructor
    ConstructorClass(int a1, int b1){
        System.out.println("Inside parameterized constructor");
        a=a1;
        b=b1;
    }
}
public class Main{
    public static void main(String[] args) {
        ConstructorClass obj = new ConstructorClass();
        System.out.println(obj.a);
        System.out.println(obj.b);
    }
}
```

OUTPUT



```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Inside parameterized constructor
Inside default constructor
10
5

C:\Users\Dell\Desktop>
```

MODULE 2 – Q9

CODE

```
public class simpleGeometricObject {
    private String color = "White";
    private boolean filled;
    private java.util.Date dateCreated;
    public simpleGeometricObject(){
        dateCreated = new java.util.Date();
    }
    public String getColor(){
        return color;
    }
    public void setColor(String color){
        this.color = color;
    }
    public boolean isFilled(){
        return filled;
    }
    public void setFilled(boolean filled){
        this.filled = filled;
    }
    public java.util.Date getDateCreated(){
        return dateCreated;
    }
    void disp(){
        System.out.println("Object created on: " + dateCreated);
        System.out.println("Color of the Object: " + color);
        System.out.println("Status_Filled: " + filled);
    }
}
public class circle extends simpleGeometricObject{
    private double radius;
    public circle(){}
    public circle(double radius){
        this.radius = radius;
    }
    public circle(double radius, String color, boolean filled){
        this.radius = radius;
        setColor(color);
        setFilled(filled);
    }
    public double getRadius(){
        return radius;
    }
    public void setRadius(double radius){
        this.radius = radius;
    }
    public double getArea(){
        return radius*radius*Math.PI;
    }
    public double getPerimeter(){
        return 2*radius*Math.PI;
    }
    public double getDiameter(){
        return 2*radius;
    }
    void disp(){
        super.disp();
        System.out.println("Radius of circle: " + radius);
        System.out.println("Diameter of circle: " + getDiameter());
        System.out.println("Perimeter of circle: " + getPerimeter());
        System.out.println("Area of circle: " + getArea());
    }
}

import java.util.Scanner;
public class testGeometricObjects {
    public static void main(String [] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter radius of circle 1: ");
        double radius1 = sc.nextDouble();
        System.out.print("Enter color of circle 1: ");
        String color1 = sc.next();
        System.out.print("Is the circle filled? ");
        boolean filled1 = sc.nextBoolean();
        System.out.print("Enter radius of circle 2: ");
        double radius2 = sc.nextDouble();
        System.out.print("Enter color of circle 2: ");
        String color2 = sc.next();
        System.out.print("Is the circle filled? ");
        boolean filled2 = sc.nextBoolean();
        circle c1 = new circle(radius1, color1, filled1); //Parametrized Constructor to invoke object c1
        circle c2 = new circle(); //No-arg constructor to invoke an empty object
        //setting parameters of object c2
        c2.setColor(color2);
        c2.setFilled(filled2);
        c2.setRadius(radius2);
        c1.disp(); //Calls display function for c1
        c2.disp(); //Calls display function for c2
        sc.close();
    }
}
```

OUTPUT

```
Enter radius of circle 1: 10
Enter color of circle 1: blue
Is the circle filled? True
Enter radius of circle 2: 20
Enter color of circle 2: black
Is the circle filled? False
Object created on: Fri Sep 16 17:29:00 IST 2022
Color of the Object: blue
Status_Filled: true
Radius of circle: 10.0
Diameter of circle: 20.0
Perimeter of circle: 62.83185307179586
Area of circle: 314.1592653589793
Object created on: Fri Sep 16 17:29:00 IST 2022
Color of the Object: black
Status_Filled: false
Radius of circle: 20.0
Diameter of circle: 40.0
Perimeter of circle: 125.66370614359172
Area of circle: 1256.6370614359173
```

MODULE 2 – Q10, Q11

CODE

Code 1: student.java

```
public class student {
    private int age;
    private String name;
    public student(){}
    public student(int age, String name){
        this.age = age;
        this.name = name;
    }
    public String getName(){
        return name;
    }
    public int getAge(){
        return age;
    }
    public void setAge(int age){
```

```java
        this.age = age;
    }
    public void setName(String name){
        this.name = name;
    }
    void disp(){
        System.out.println("Name of Student: " + name);
        System.out.println("Age of Student: " + age);
    }
}
```

Code 2: engineeringStudent.java

```java
public class engineeringStudent extends student{
    private String rollnumber;
    private double cgpa;
    public engineeringStudent(){}
    public engineeringStudent(String rollnumber, double cgpa){
        this.rollnumber = rollnumber;
        this.cgpa = cgpa;
    }
    public String getrollNumber(){
        return rollnumber;
    }
    public double getCGPA(){
        return cgpa;
    }
    public void setCGPA(double cgpa){
        this.cgpa = cgpa;
    }
    public void setRollNumber(String rollnumber){
        this.rollnumber = rollnumber;
    }
    void disp(){
        super.disp();
        System.out.println("Roll Number: " + rollnumber);
        System.out.println("CGPA: " + cgpa);
    }
}
```

Code 3: highschoolstudent.java

```java
public class highschoolstudent extends student{
    private double physmarks;
    private double mathmarks;
    private double chemmarks;
    highschoolstudent(){}
    highschoolstudent(double physmarks, double mathmarks, double chemmarks){
        this.chemmarks = chemmarks;
        this.mathmarks = mathmarks;
        this.physmarks = physmarks;
    }
    public double getphysmarks(){
        return physmarks;
    }
    public double getmathmarks(){
        return mathmarks;
    }
    public double getchemmarks(){
        return chemmarks;
    }
    public void setphysmarks(double physmarks){
        this.physmarks = physmarks;
    }
    public void setchemmarks(double chemmarks){
        this.chemmarks = chemmarks;
    }
    public void setmathmarks(double mathmarks){
        this.mathmarks = mathmarks;
    }
    void disp(){
        super.disp();
        System.out.println("Physics Marks: " + physmarks);
        System.out.println("Chemistry Marks: " + chemmarks);
        System.out.println("Mathematics Marks: " + mathmarks);
    }
}
```

Code 4: cseStudent.java

```java
public class cseStudent extends engineeringStudent{
    private String language;
    private String projectname;
    cseStudent(){}
    cseStudent(String language, String projectname){
        this.language = language;
        this.projectname = projectname;
    }
    public String getLanguage(){
        return language;
    }
    public String getprojectname(){
        return projectname;
    }
    public void setlanguage(String language){
        this.language = language;
    }
    public void setproject(String projectname){
        this.projectname = projectname;
    }
    void disp(){
        super.disp();
        System.out.println("Language: " + language);
        System.out.println("Project Name: " + projectname);
    }
}
```

Code 5: MultiInheritance.java

```
public class MultiInheritance {
    public static void main(String [] args) {
        highschoolstudent s1 = new highschoolstudent(); //High School Student
        s1.setAge(18);
        s1.setName("ABC");
        s1.setchemmarks(69);
        s1.setmathmarks(89);
        s1.setphysmarks(78);
        s1.disp();
        cseStudent s2 = new cseStudent(); //CSE Student
        s2.setAge(19);
        s2.setName("XYZ");
        s2.setCGPA(9.89);
        s2.setRollNumber("21BCP050");
        s2.setlanguage("Java");
        s2.setproject("Project 1");
        s2.disp();
    }
}
```

OUTPUT

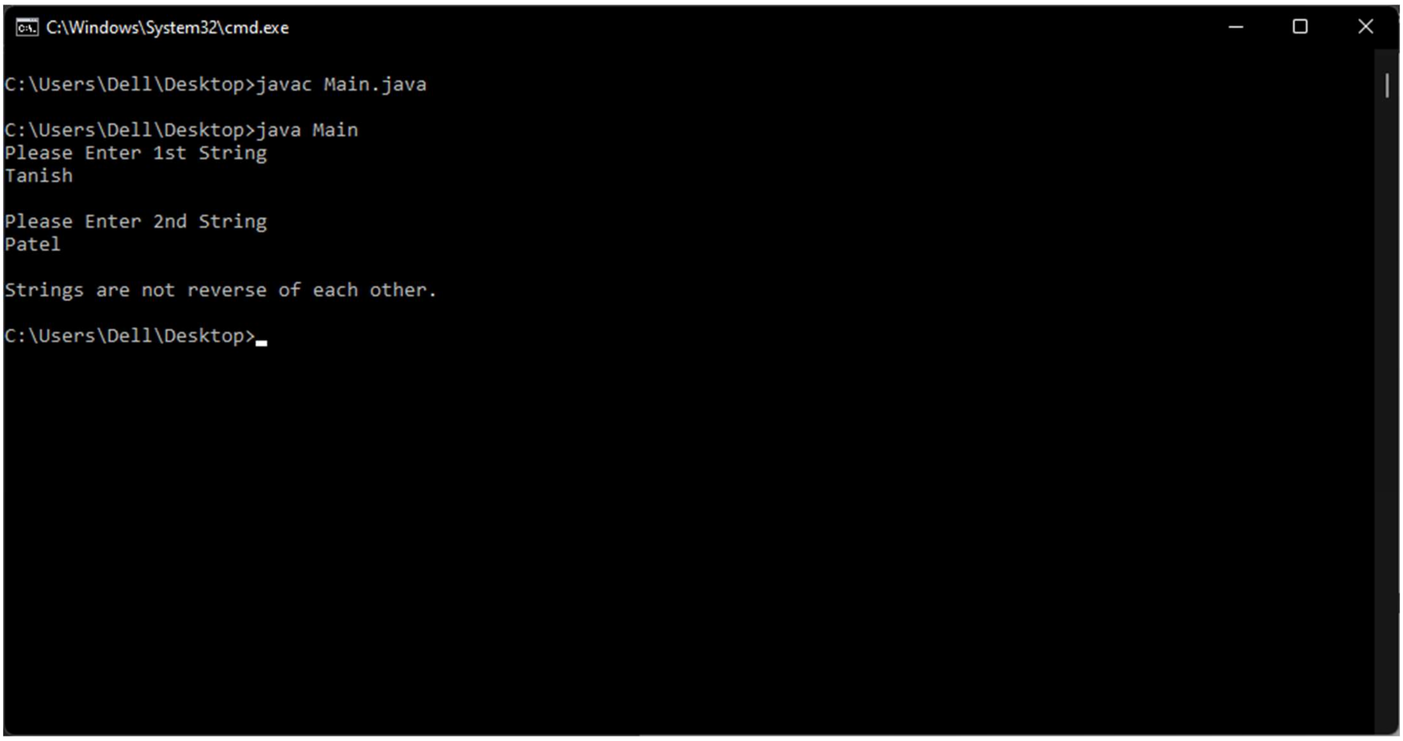
```
Name of Student: ABC
Age of Student: 18
Physics Marks: 78.0
Chemistry Marks: 69.0
Mathematics Marks: 89.0
Name of Student: XYZ
Age of Student: 19
Roll Number: 21BCP050
CGPA: 9.89
Language: Java
Project Name: Project 1
```

MODULE 2 – Q12

CODE

```
import java.lang.StringBuilder;
import java.util.*;
public class Main{
    static void reverseCheck(String s1, String s2){
        if (s1.length()!=s2.length()){
            System.out.println("Strings are not reverse of each other.");
        }
        else{
            String temp="";
            for (int i=s1.length() - 1;i>=0;i--){
                temp+=s1.charAt(i);
            }
            for (int i=0;i<s1.length();i++){
                if (temp.charAt(i)!=s2.charAt(i)){
                    System.out.println("Strings are not reverse of each other.");
                    return;
                }
            }
            System.out.println("Strings are reverse of each other.");
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Please Enter 1st String");
        String a = sc.nextLine();

        System.out.println("\nPlease Enter 2nd String");
        String b = sc.nextLine();

        System.out.println();
        reverseCheck(a,b);
    }
}
```

OUTPUT



```
C:\Users\Dell\Desktop>javac Main.java

C:\Users\Dell\Desktop>java Main
Please Enter 1st String
Tanish

Please Enter 2nd String
Patel

Strings are not reverse of each other.

C:\Users\Dell\Desktop>_
```

MODULE 2 – Q13, Q14

CODE

Code: geometricObjects.java

```
abstract class geometricObjects {
    private String color;
    private boolean filled;
    protected geometricObjects() {}
    protected geometricObjects(String color, boolean filled) {
        this.color = color;
        this.filled = filled;
    }
    protected geometricObjects(boolean filled) {
        this.filled = filled;
    }
    protected geometricObjects(String color) {
        this.color = color;
    }
    abstract double getArea();
    abstract double getPerimeter();
    protected String getColor() {
        return color;
    }
    protected void setColor(String color) {
        this.color = color;
    }
    protected boolean isFilled() {
        return filled;
    }
    protected void setFilled(boolean filled) {
        this.filled = filled;
    }
    protected void disp(){
        System.out.println("Color: "+color);
        System.out.println("Filled: "+filled);
    }
}
```

Code: rectangle.java

```
public class rectangle extends geometricObjects{
    private double width;
    private double height;
    public rectangle() {}
    public rectangle(double width, double height) {
        this.width = width;
        this.height = height;
    }
    public rectangle(double width, double height, String color, boolean filled) {
        super(color, filled);
        this.width = width;
        this.height = height;
    }
    public double getWidth() {
        return width;
    }
    public void setWidth(double width) {
        this.width = width;
    }
    public double getHeight() {
        return height;
    }
    public void setHeight(double height) {
        this.height = height;
    }
    //Defining the abstract methods
    public double getArea() {
        return width*height;
    }
    public double getPerimeter() {
        return 2*(width+height);
    }
    public void disp(){
        super.disp();
        System.out.println("Width: "+width);
        System.out.println("Height: "+height);
        System.out.println("Area: "+getArea());
        System.out.println("Perimeter: "+getPerimeter());
    }
}
```

Code: circle.java

```
public class circle extends geometricObjects{
    private double radius;
    public circle() {}
    public circle(double radius) {
        this.radius = radius;
    }
    public circle(double radius, String color, boolean filled) {
        super(color, filled);
        this.radius = radius;
    }
    public double getRadius() {
        return radius;
    }
    public void setRadius(double radius) {
        this.radius = radius;
    }
    //Defining the abstract methods
    public double getArea() {
        return radius*radius*Math.PI;
    }
    public double getPerimeter() {
        return 2*radius*Math.PI;
    }
    public void disp(){
        super.disp();
        System.out.println("Radius: "+radius);
        System.out.println("Area: "+getArea());
        System.out.println("Perimeter: "+getPerimeter());
    }
}
```

Code: triangle.java

```
public class triangle extends geometricObjects{
    private double side1;
    private double side2;
    private double side3;
    public triangle() {}
    public triangle(double side1, double side2, double side3) {
```

```java
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }
    public triangle(double side1, double side2, double side3, String color, boolean filled) {
        super(color, filled);
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }
    public double getSide1() {
        return side1;
    }
    public void setSide1(double side1) {
        this.side1 = side1;
    }
    public double getSide2() {
        return side2;
    }
    public void setSide2(double side2) {
        this.side2 = side2;
    }
    public double getSide3() {
        return side3;
    }
    public void setSide3(double side3) {
        this.side3 = side3;
    }
    //Defining the abstract methods
    public double getArea() {
        double s = (side1+side2+side3)/2;
        return Math.sqrt(s*(s-side1)*(s-side2)*(s-side3));
    }
    public double getPerimeter() {
        return side1+side2+side3;
    }
    public void disp(){
        super.disp();
        System.out.println("Side1: "+side1);
        System.out.println("Side2: "+side2);
        System.out.println("Side3: "+side3);
        System.out.println("Area: "+getArea());
        System.out.println("Perimeter: "+getPerimeter());
    }
}
```

Code: testObjects.java

```java
import java.util.Scanner;
public class testObjects {
    public static void main(String [] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("CIRCLE");
        System.out.println("Enter the radius of the circle: ");
        double radius = input.nextDouble();
        System.out.println("Enter the color of the circle: ");
        String color = input.next();
        System.out.println("Is the circle filled? (true/false): ");
        boolean filled = input.nextBoolean();
        circle c = new circle(radius, color, filled);
        System.out.println();
        System.out.println("RECTANGLE");
        System.out.println("Enter the width of the rectangle: ");
        double width = input.nextDouble();
        System.out.println("Enter the height of the rectangle: ");
        double height = input.nextDouble();
        System.out.println("Enter the color of the rectangle: ");
        color = input.next();
        System.out.println("Is the rectangle filled? (true/false): ");
        filled = input.nextBoolean();
        rectangle r = new rectangle(width, height, color, filled);
        System.out.println();
        System.out.println("TRIANGLE");
        System.out.println("Enter the side 1 of triangle: ");
        double side1 = input.nextDouble();
        System.out.println("Enter the side 2 of triangle: ");
        double side2 = input.nextDouble();
        System.out.println("Enter the side 3 of triangle: ");
        double side3 = input.nextDouble();
        System.out.println("Enter the color of the triangle: ");
        color = input.next();
        System.out.println("Is the triangle filled? (true/false): ");
        filled = input.nextBoolean();
        triangle t = new triangle(side1, side2, side3, color, filled);
        System.out.println();
        System.out.println("Circle: ");
        c.disp();
        System.out.println();
        System.out.println("Rectangle: ");
        r.disp();
        System.out.println();
        System.out.println("Triangle: ");
        t.disp();
    }
}
```

OUTPUT

```
CIRCLE
Enter the radius of the circle:
10
Enter the color of the circle:
blue
Is the circle filled? (true/false):
true
RECTANGLE
```

```
Enter the width of the rectangle:
69
Enter the height of the rectangle:
96
Enter the color of the rectangle:
red
Is the rectangle filled? (true/false):
false
TRIANGLE
Enter the side 1 of triangle:
30
Enter the side 2 of triangle:
40
Enter the side 3 of triangle:
50
Enter the color of the triangle:
brown
Is the triangle filled? (true/false):
True

Circle:
Color: blue
Filled: true
Radius: 10.0
Area: 314.1592653589793
Perimeter: 62.83185307179586
Rectangle:
Color: red
Filled: false
Width: 69.0
Height: 96.0
Area: 6624.0
Perimeter: 330.0
Triangle:
Color: brown
Filled: true
Side1: 30.0
Side2: 40.0
Side3: 50.0
Area: 600.0
Perimeter: 120.0
```