Practical 10 :

Code :

class add {

    int no1;

    int no2;

    add() {

        System.out.println("Constructor is called.");

    }

    add(int a, int b) {

        this.no1 = a;

        this.no2 = b;

    }

}

class Multi\_const {

    public static void main(String[] args) {

        System.out.println("Default Constructor :");

        new add();

        System.out.println("Paramieterised Constructor : ");

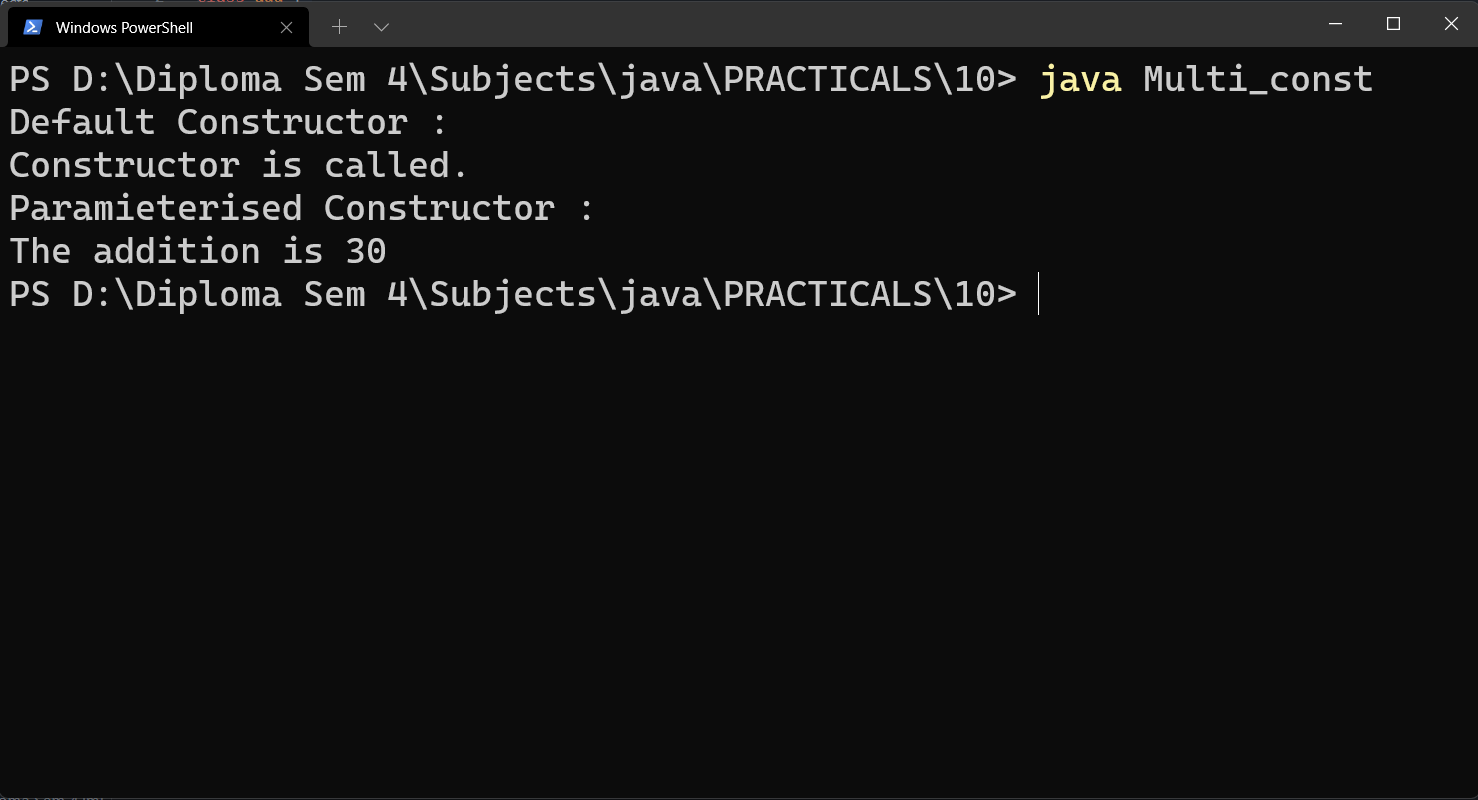
        add ad = new add(20, 10);

        int c = ad.no1 + ad.no2;

        System.out.println("The addition is " + c);  }

}

Output :



Code :

public class Complex {

    int real;

    int img;

    Complex(int r,int i){

        this.real=r;

        this.img=i;

    }

    static void sum(Complex n1,Complex n2){

        Complex res = new Complex(0, 0);

        res.real = n1.real+n2.real;

        res.img = n1.img+n2.img;

        System.out.println(" "+n1.real+" "+n1.img+" i\n+");

        System.out.println(" "+n1.real+" "+n1.img+" i");

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

        System.out.println(" "+res.real+" "+res.img+" i");

    }

    public static void main(String[] args) {

        Complex c1 = new Complex(10, 20);

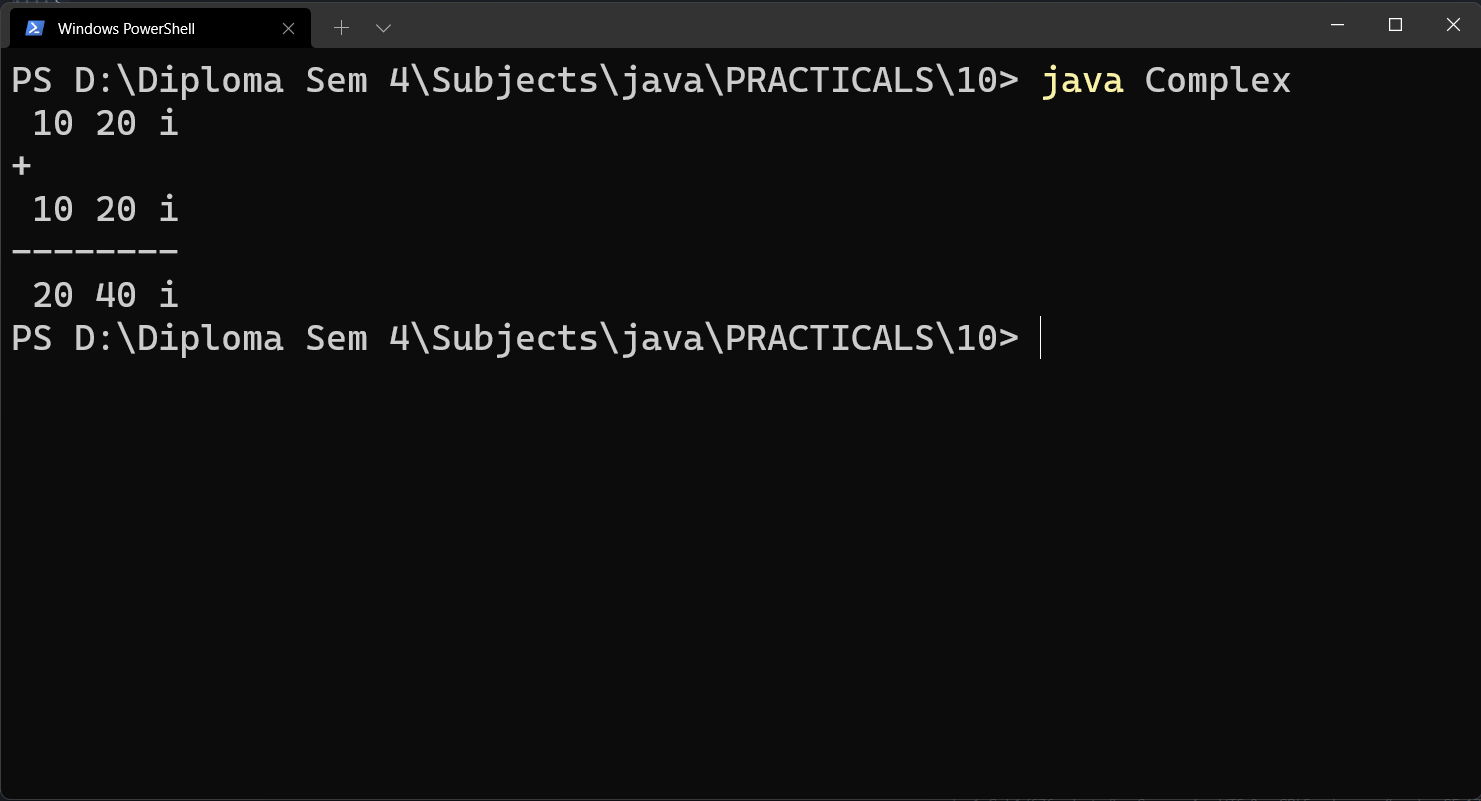
        Complex c2 = new Complex(10, 20);

        sum(c1,c2);

    }

}

Output :



Code :

public class Point {

    int m\_x,m\_y;

    public Point(){}

    public Point(int x , int y){

        m\_x=x;

        m\_y=y;

    }

    public static void main(String[] args) {

        Point p1 =  new Point();

        Point p =  new Point(2,3);

        System.out.println("x"+p.m\_x);

        System.out.println("y"+p.m\_y);

        System.out.println("y"+p1.m\_y);

        System.out.println("y"+p1.m\_y);

    }

}

Output :

