Assignment – 4

Codes

1.) public class Static {

    private String name = "Static class";

    public static void first(){}

    public static void second(){}

    public void third(){

        System.out.println(name);

    }

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

    first();

    second();

    Static s = new Static();

    s.third();

    }

}

2.) public class A4Q2 {

}

class calltest{

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

        String s=new String("English");

        fun(s);

        System.out.println(s);

    }

    static String fun(String *s*){

        System.out.println(*s*);

*s*="Hindi";

        System.out.println(*s*);

        return *s*;

    }

}

class intc{

    int x;

}

class reftest{

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

        intc o=new intc();

        fun(o);

        System.out.println(o.x);

    }

    static void fun(*intc* *s*){

        System.out.println(*s*.x);

*s*.x=10;

        System.out.println(*s*);

    }

}

3.) public class A4Q3 {

}

class Percolate{

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

        int[] dataSeq = {6,4,8,2,1};

        printIntArray(dataSeq);

        for (int index = 1; index <dataSeq.length; ++index)

            if (dataSeq[index-1] >dataSeq[index])

                swap(dataSeq, index-1, index);

        printIntArray(dataSeq);

    }

    public static void swap(int[] *intArray*, int *i*, int *j*) {

        int tmp = *intArray*[*i*];

*intArray*[*i*] = *intArray*[*j*];

*intArray*[*j*] = tmp;

    }

    public static void swap(int *v1*, int *v2*) {

        int tmp = *v1*;

*v1* = *v2*;

*v2* = tmp;

    }

    public static void printIntArray(int[] *array*) {

        for (int value : *array*)

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

        System.out.println();

    }

}

4.) public class A4Q4 {

}

class ParameterPass{

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

        int i = 0;

        addTwo(i++);

        System.out.println(i);

    }

    static void addTwo(int *i*) {

*i* += 2;

    }

}

5.) import java.util.Scanner;

public class A4Q5{

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

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

        Scanner scan = new Scanner(System.in);

        long number = scan.nextLong();

        System.out.println("Enter the number of digit you want to know : ");

        int k = scan.nextInt();

        System.out.println(digit(number, k));

    }

    static int digit(long *number*, int *k*){

        int ret;

        long numberCopy = *number*;

        int  count =0 ;

        while(*number*!=0){

*number*/=10;

            count++;

        }

        int div = count-*k*;

        numberCopy/=Math.pow(10, div);

        if(numberCopy>0)

            return (int) numberCopy%10;

        else{

            return 0;

        }

    }

}

6.) import java.util.Scanner;

public class A4Q6{

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

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter the number of triangular number : ");

        int index = scan.nextInt();

        System.out.println(index+"th triangular number is : "+t(index));

    }

    static long t(int *n*){

        if(*n*==1)

            return 1;

        else{

            return *n*+t(*n*-1);

        }

    }

}

7.) public class A4Q7{

    static int j;

    static{

        j=10;

        method1();

    }

    static void method1(){

        System.out.println("static block 1 called !"+'\n'+"value of j is : "+j);

    }

    static void method2(){

        System.out.println("static block 2 called !"+'\n'+"value of j is : "+j);

    }

    void method3(){

        System.out.println("called from main method !");

    }

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

        A4Q7 a = new A4Q7();

        a.method3();

    }

    static{

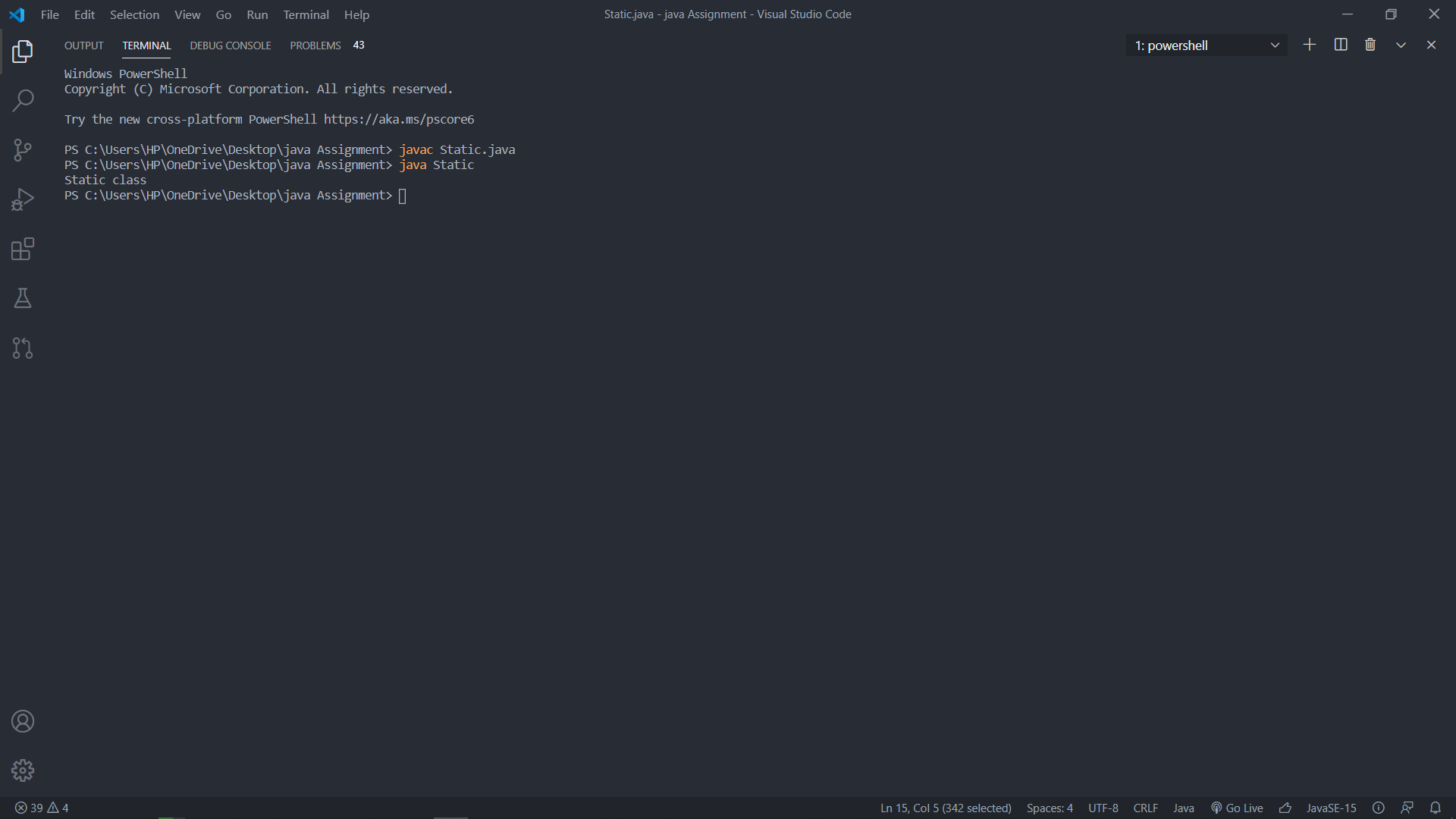
        j=30;

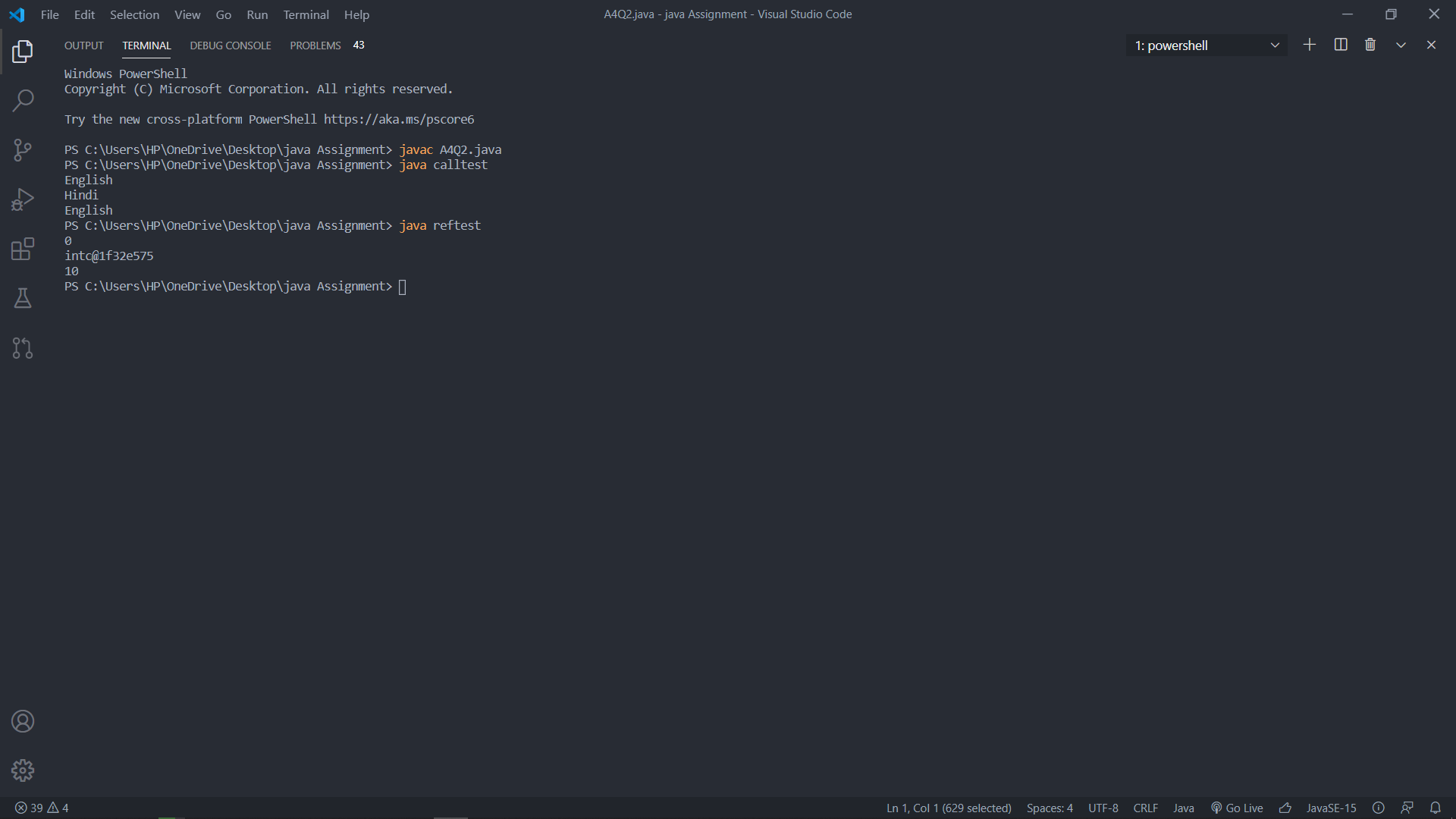
        method2();

    }

}

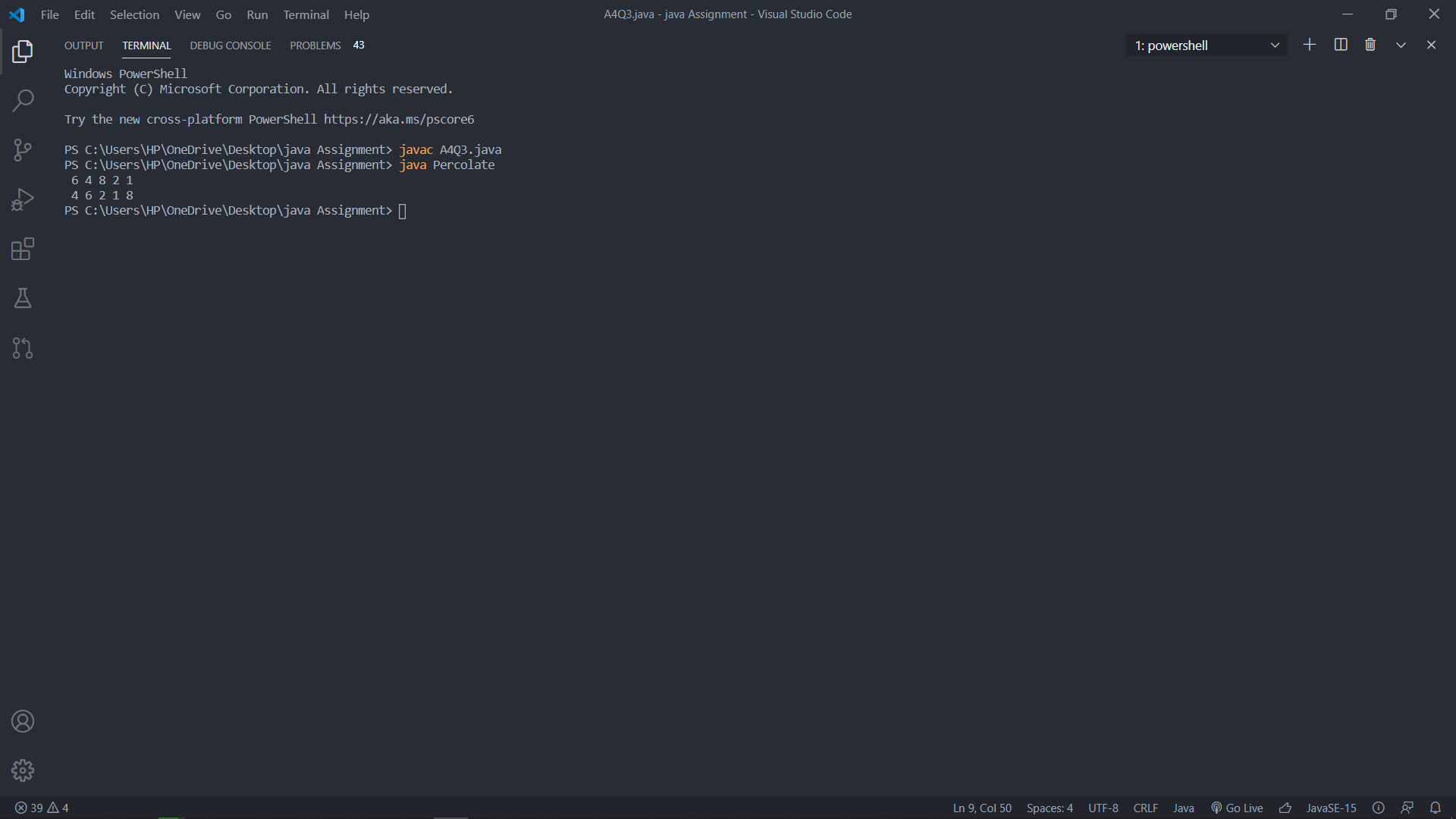
Outputs

1.) 

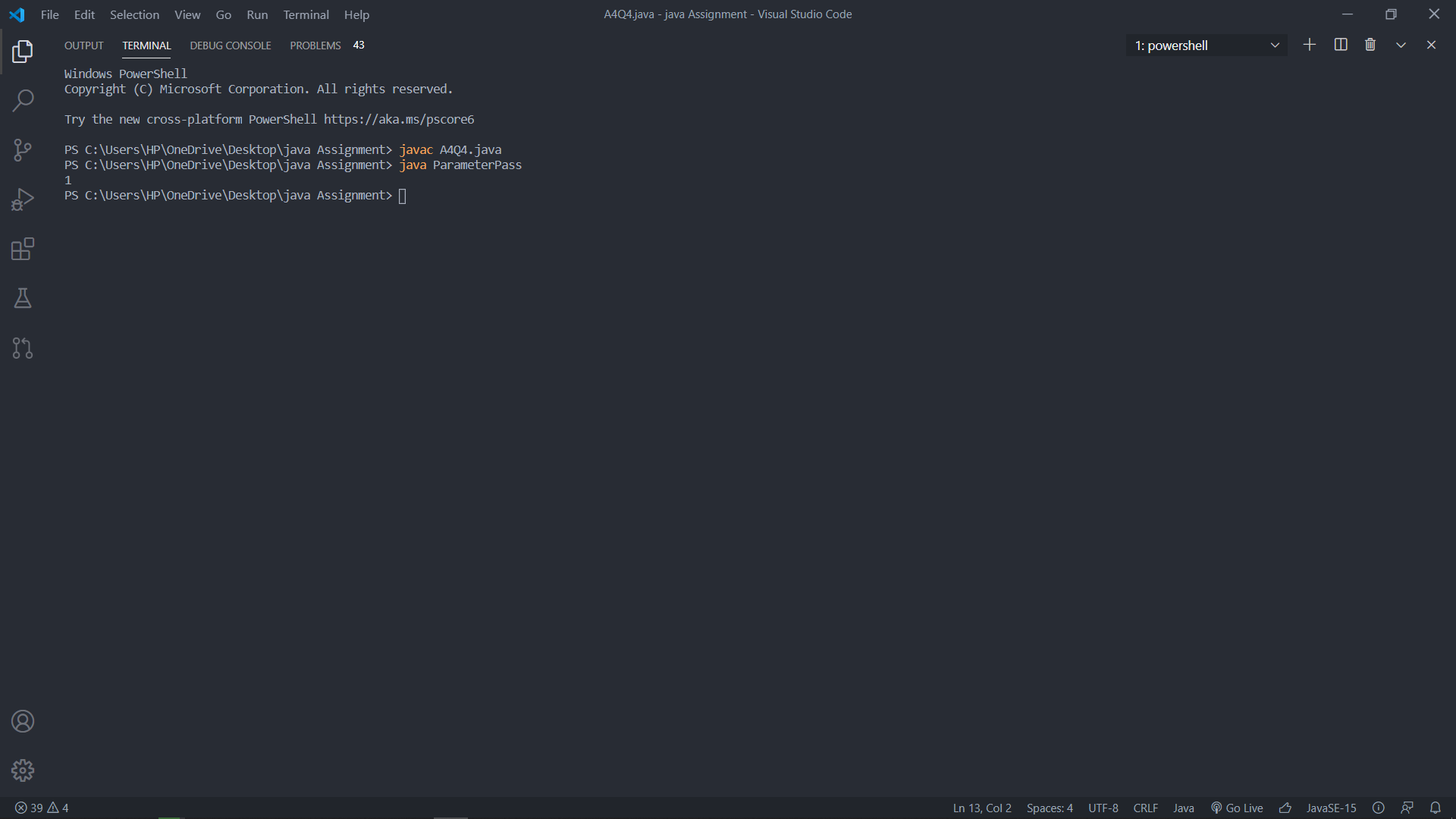
2.) 

**Explanation:** Java passes the parameter by value only.

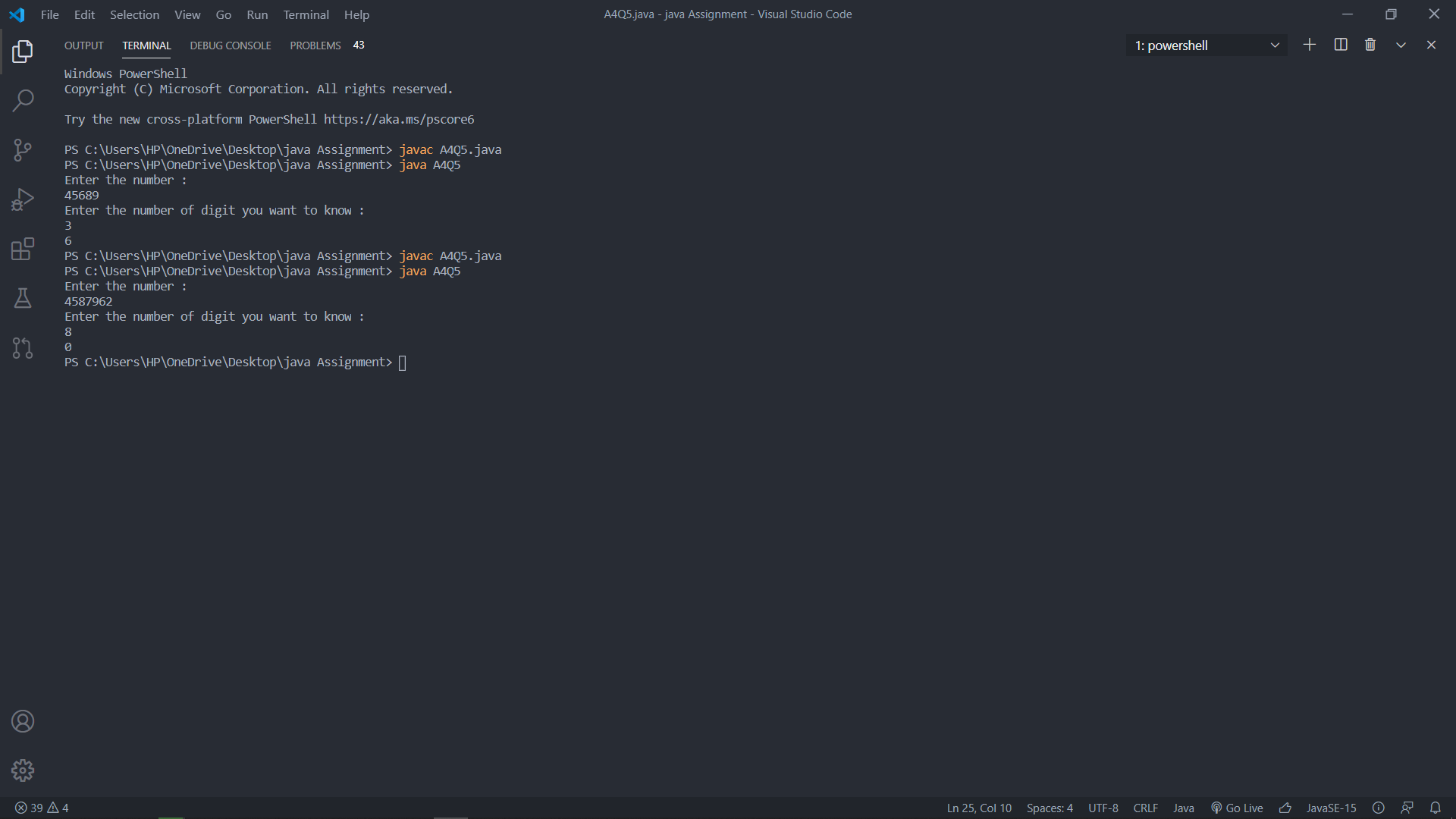
1. In the first class i.e. “calltest” the string’s reference is passed as a value in the “fun” method, which creates a copy of the reference and it points to “Hindi”. But when the control switches over to the main method again the value of string is the same i.e. “English”. It does not change the reference but the copy of reference is updated.
2. In the second class the reference of the object is passed as a value to the function which in turn points to the same object reference in the heap store. Therefore the change made in the copy reflects in the original object also.

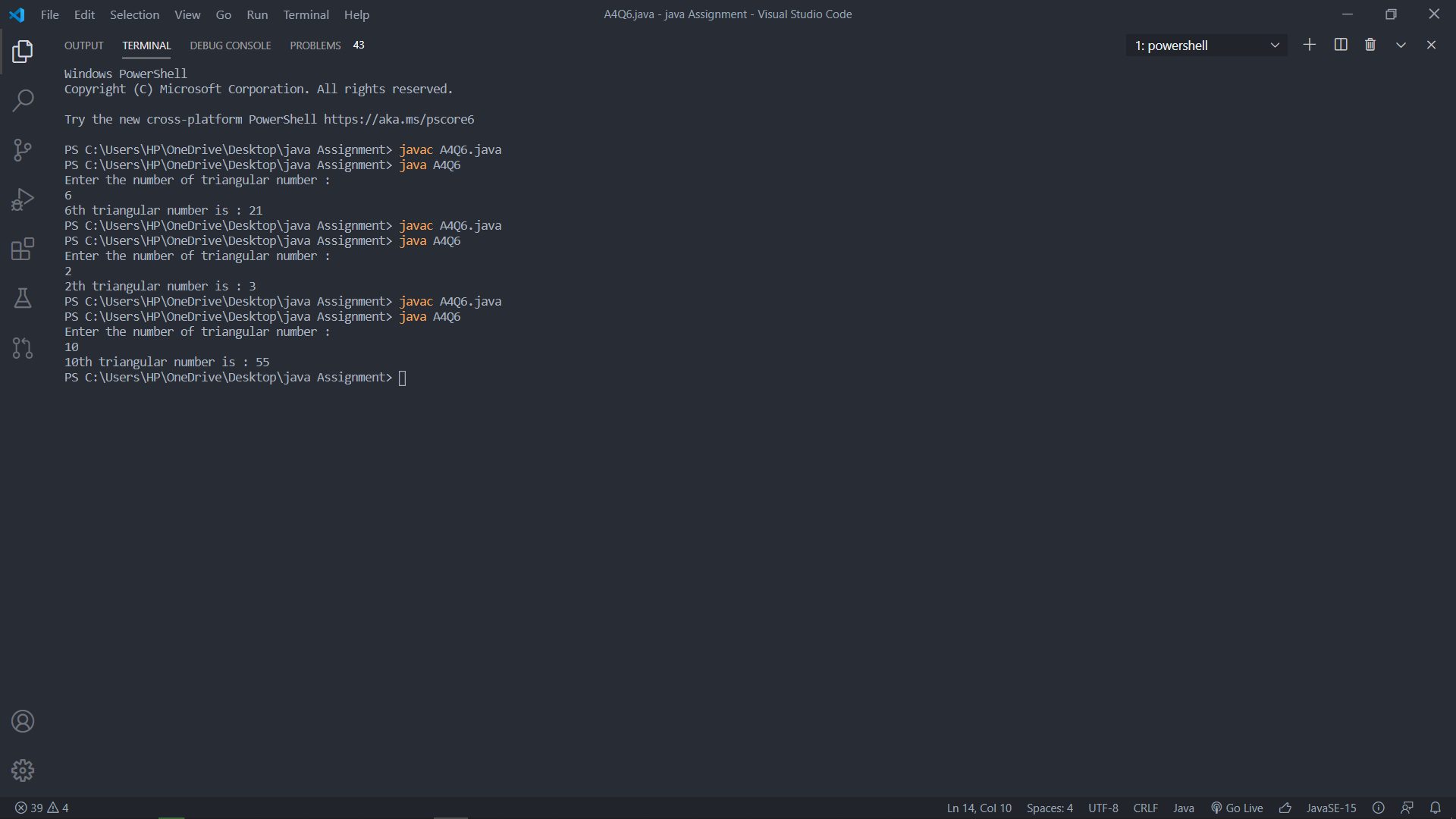
3.) 

**Explanation:** The arrays in java are passed as a reference value in the form of parameters. Therefore the changes made in the method to the copy of array (which in turn points to the same memory location) reflects in the original array a well.

4.) 

**Explanation:** The post increment operator passes the value first and then the increment is performed. Therefore when “i++” is passed in the method “0” is passed and not “1”. But again since it is not an object so copy of reference is created and it points to”i+=2”, and the value originally in “i” is not changed.

5.) 

6.) 

7.) 