Assignment No. => 7

Codes

1.) package myFirstPackage;

import java.util.Scanner;

class bmiCalculator{

    public double height, weight;

    public double getData(){

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter weight in kgs ");

        weight = scan.nextDouble();

        System.out.println("Enter height in meters ");

        height = scan.nextDouble();

        return bmi(weight,height);

    }

    static double bmi(double wt, double ht){

        return wt/(ht\*ht);

    }

    public double getwt(){

        return weight;

    }

    public double getht(){

        return height;

    }

}

class displayData{

    public void display(double wt, double ht, double bmi){

        Scanner scan = new Scanner(System.in);

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

        String name = scan.nextLine();

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

        int age = scan.nextInt();

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

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

        System.out.println("BMI : "+bmi);

        if(bmi<18){

            System.out.println("You are underweight. Eat more!");

        }

        if(bmi>18 && bmi<=25){

            System.out.println("You are fit. Keep it up!");

        }

        if(bmi>25 && bmi<=30){

            System.out.println("You are overeight. Lose some flesh!");

        }

        if(bmi>30){

            System.out.println("You are obese. Not good!");

        }

    }

}

public class A7Q1 {

    public static void main(String args[]){

        A7Q1 a = new A7Q1();

        a.use();

    }

    public void use(){

        bmiCalculator b = new bmiCalculator();

        displayData d = new displayData();

        d.display(b.getwt(), b.getht(), b.getData());

        System.out.println(b.getht());

    }

}

2.) class MyException extends Exception{

    private static final long serialVersionUID = 1L;

    MyException(String s) {

        super(s);

    }

}

public class A7Q2 {

    static void validate(String a) throws MyException{

        if(a.equals("Aditya"))

            throw new MyException("match");

        else

            System.out.println("no match");

    }

    public static void main(String args[]){

        try {

            validate("Aditya");

            System.out.println("Since your name is not aditya yo get 18 marks in all MSTs of data structure");

        }

        catch (Exception e) {

            System.out.println("since your name is aditya you get 14 marks in all MSTs of data structure");

        }

    }

}

3.) public class A7Q3 {

    public static void main(String arsg[]){

        int z;

        try{

            z=10/0;

        }

        catch(Exception e){

            System.out.println("caught" + e.getMessage());

        }

        catch(ArithmeticException e){

            System.out.println("caught" + e.getMessage());

        }

    }

}

**Unreachable catch block:**  As the name suggests the catch block which is not in the reach of the java compiler is called unreachable catch block and the error aroused due to it is called unreachable catch block error.

In the above program, ArithmeticException is the child class of Exception class, therefore when we write Exception in first catch block it gets caught and does not go to the second catch block, although the exception we throw is the ArithmeticException.

This is called Unreachable catch block error, here the catch block with ArithmeticException is not reachable.

4.) public class A7Q4{

    static void demo(){

        try {

            throw new NullPointerException();

        }

        catch (NullPointerException e) {

            System.out.println("caught "+e);

            System.out.println("caught first time");

            throw new ArithmeticException();

        }

    }

    public static void main(String args[]){

        try{

            demo();

        }

        catch(ArithmeticException e){

            System.out.println("caught "+e);

            System.out.println("caught second time");

        }

    }

}

**Re-Throwing an Exception:** Sometimes we may need to re-throw an exception in Java. If a catch block cannot handle the particular exception it has caught, we can re-throw the exception. The re-throw expression causes the **originally thrown object to be re-thrown.**

**Here in the above example we call “demo()” function which has a try catch block and it throws NullPointerException, and the catch block catches it and again throw ArithmeticException, which in turn is caught by the catch block in the main method. Here we threw the exception explicitly, this concept is called re-throwing exception.**

5.) import java.io.\*;

import java.util.Scanner;

public class A7Q5 {

    public static void main(String args[]) throws IOException{

        Scanner scan = new Scanner(System.in);

        System.out.println("enter the file from which to copy content : ");

        String fileFromCopyContent = scan.nextLine();

        System.out.println("enter the file to which copy the content : ");

        String fileToCopyContent = scan.nextLine();

        FileReader r = new FileReader(fileFromCopyContent);

        FileWriter w = new FileWriter(fileToCopyContent);

        BufferedReader br = new BufferedReader(r);

        BufferedWriter bw = new BufferedWriter(w);

        int i;

        while((i=br.read())!=-1){

            bw.write(i);

        }

        scan.close();

        br.close();

        bw.close();

        w.close();

        r.close();

    }

}

6.) import java.io.\*;

import java.util.Scanner;

public class A7Q6 {

    public static void main(String args[]) throws IOException{

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter file name / path to delete : ");

        String path = scan.nextLine();

        File f = new File(path);

        if(f.exists()){

            boolean result = f.delete();

            if(result)

                System.out.println("Successfully deleted file : " + f.getName());

            else{

                System.out.println("Failed to delete");

            }

        }

        else{

            System.out.println("File not present. ");

        }

        scan.close();

    }

}

7.) import java.io.\*;

import java.util.Scanner;

public class A7Q7 {

    public static void main(String args[]) throws IOException{

        Scanner scan = new Scanner(System.in);

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

        String file = scan.nextLine();

        File f = new File(file);

        System.out.println("File path : "+f.getAbsolutePath());

        System.out.println("File size in bits : "+f.length());

        System.out.println("File size in bytes : "+f.length()/8);

        scan.close();

    }

}

8.) import java.io.\*;

import java.util.Scanner;

public class A7Q8 {

    public static void main(String args[]) throws IOException{

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter the file path : ");

        String pathFile = scan.nextLine();

        BufferedReader b = new BufferedReader(new FileReader(pathFile));

        int i;

        while((i=b.read())!=-1){

            System.out.print((char)i);

        }

        scan.close();

        b.close();

    }

}

9.) import java.io.\*;

import java.util.Scanner;

public class A7Q9 {

    public static void main(String args[]) throws IOException{

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter file name to create or to write in :");

        String pathName = scan.nextLine();

        FileWriter f = new FileWriter(pathName);

        String words;

        BufferedWriter b = new BufferedWriter(f);

        String flag = "y";

        while(flag.equalsIgnoreCase("y")){

            System.out.println("Enter text to write : ");

            words = scan.nextLine();

            b.write(words,0,words.length());

            System.out.println("new line?");

            flag = scan.nextLine();

            if(flag.equalsIgnoreCase("y"))

                b.newLine();

            System.out.println("More text ? : ");

            flag = scan.nextLine();

        }

        scan.close();

        b.close();

    }

}

10.) import java.io.\*;

import java.util.Scanner;

public class A7Q10 {

    public static void main(String args[]) throws IOException {

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter file to read");

        String pathFile = scan.nextLine();

        File g = new File(pathFile);

        if(g.exists()){

            FileInputStream f = new FileInputStream(g);

            int i;

            while((i=f.read())!=-1)

                System.out.print((char)i);

            f.close();

        }

        else{

            System.out.println("No such file directory found");

        }

        scan.close();

    }

}

11.) import java.io.\*;

import java.util.Scanner;

public class A7Q11 {

    public static void main(String args[]) throws IOException{

        Scanner scan = new Scanner(System.in);

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

        String path = scan.nextLine();

        FileOutputStream f = new FileOutputStream(path);

        String flag = "y";

        String words;

        while(flag.equalsIgnoreCase("y")){

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

            words = scan.nextLine();

            f.write(words.getBytes());

            System.out.println("want to write more : ");

            flag = scan.nextLine();

        }

        scan.close();

        f.close();

    }

}

12.) import java.io.\*;

import java.util.Scanner;

public class A7Q12 {

    public static void main(String args[]) throws IOException{

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter name of file : ");

        String name = scan.next();

        File f = new File(name);

        boolean result = f.createNewFile();

        if(result)

            System.out.println("File created successfully");

        else{

            System.out.println("Could not create file");

        }

        scan.close();

    }

}

13.) import java.io.\*;

import java.util.Scanner;

public class A7Q13 {

    public static void main(String args[]){

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter path to create directory without directory name : ");

        String path = scan.nextLine();

        System.out.println("Enter name of directory : ");

        String name = scan.next();

        path = path + name;

        File f = new File(path);

        boolean result = f.mkdir();

        if(result)

            System.out.println("Directory created successfully");

        else{

            System.out.println("Could not create directory");

        }

        scan.close();

    }

}

14.) import java.io.\*;

import java.util.Scanner;

public class A7Q14 {

    public static void main(String args[]) throws IOException{

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter file path or name(if in same): ");

        String file = scan.nextLine();

        FileReader f = new FileReader(file);

        BufferedReader b = new BufferedReader(f);

        int i;

        String word;

        String wordsL[];

        int space=0, tab=0, words=0, newline=0;

        while((i=b.read())!=-1){

            if((char)i==' ')

                space++;

            if((char)i=='\t')

                tab++;

            if((char)i=='\n')

                newline++;

        }

        FileReader g = new FileReader(file);

        BufferedReader d = new BufferedReader(g);

        while((word=d.readLine())!=null){

            wordsL = word.split(" ");

            words+=wordsL.length;

        }

        System.out.print("Space : "+space+'\n'+"Tab : "+tab+'\n'+"Newline : "+newline+'\n'+"Words : "+words);

        scan.close();

        b.close();

        f.close();

        g.close();

        d.close();

    }

}

15.) import java.io.\*;

import java.util.Scanner;

public class A7Q15{

    public static void main(String args[]) throws IOException {

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter file name(if in same directory) or path :");

        String file = scan.nextLine();

        FileReader f = new FileReader(file);

        BufferedReader b = new BufferedReader(f);

        String search;

        System.out.println("Enter the word you want to search : ");

        search = scan.nextLine();

        String i;

        String words[];

        int count=0;

        while((i=b.readLine())!=null){

            words = i.split(" ");

            for(String j : words)

                if(j.equals(search))

                    count++;

        }

        f.close();

        b.close();

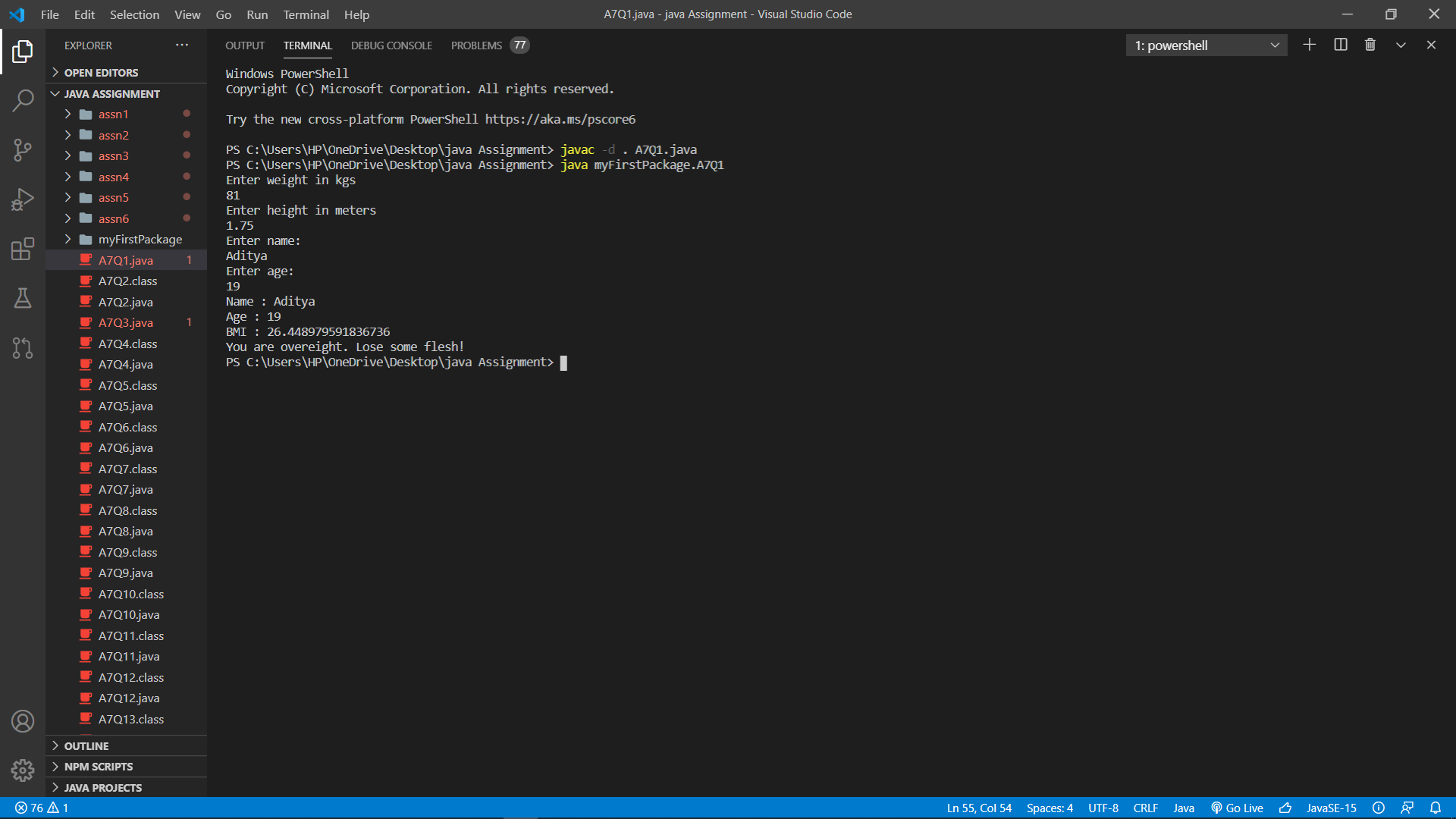
        scan.close();

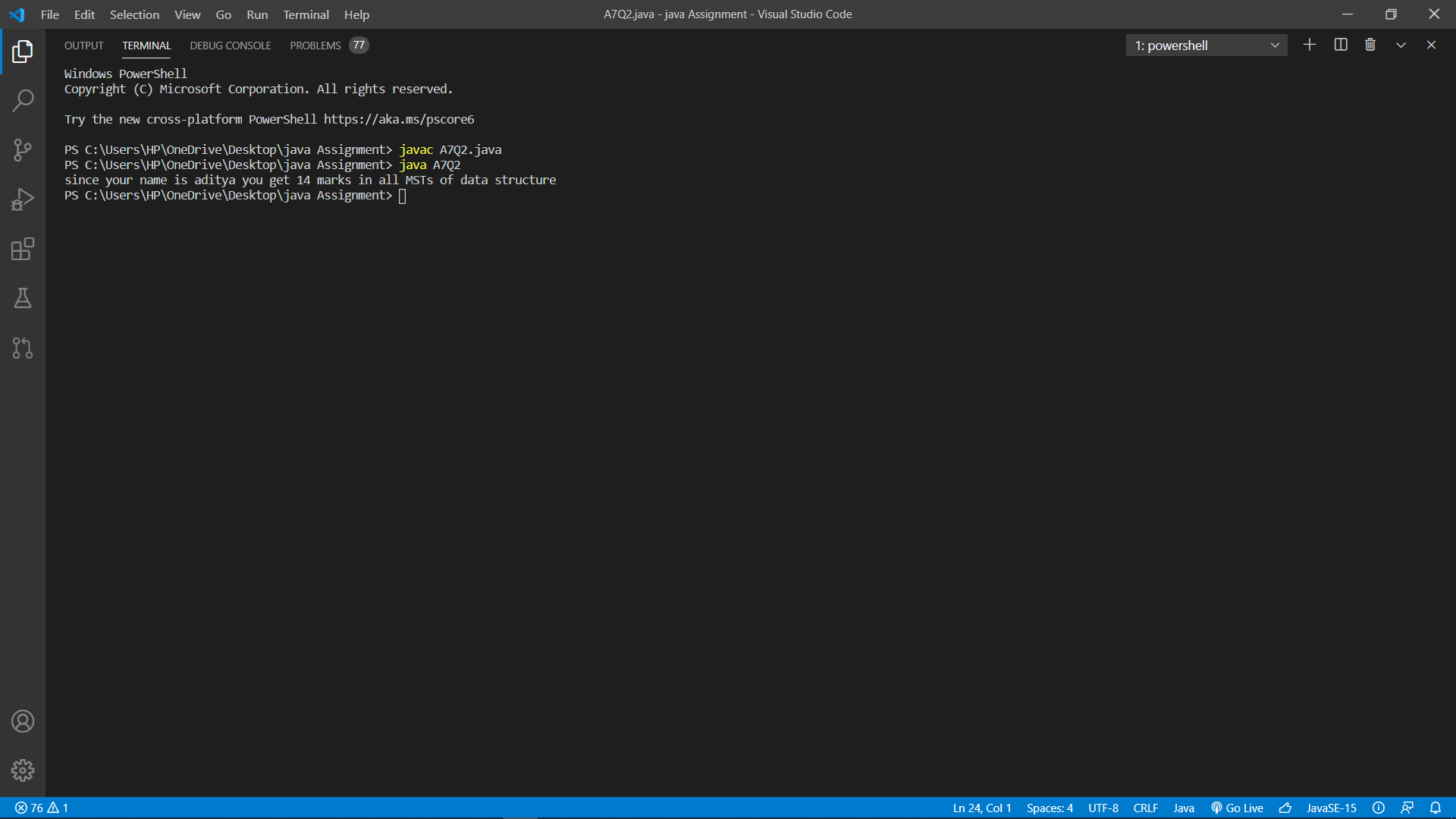
        System.out.println("appeared "+count+" times.");

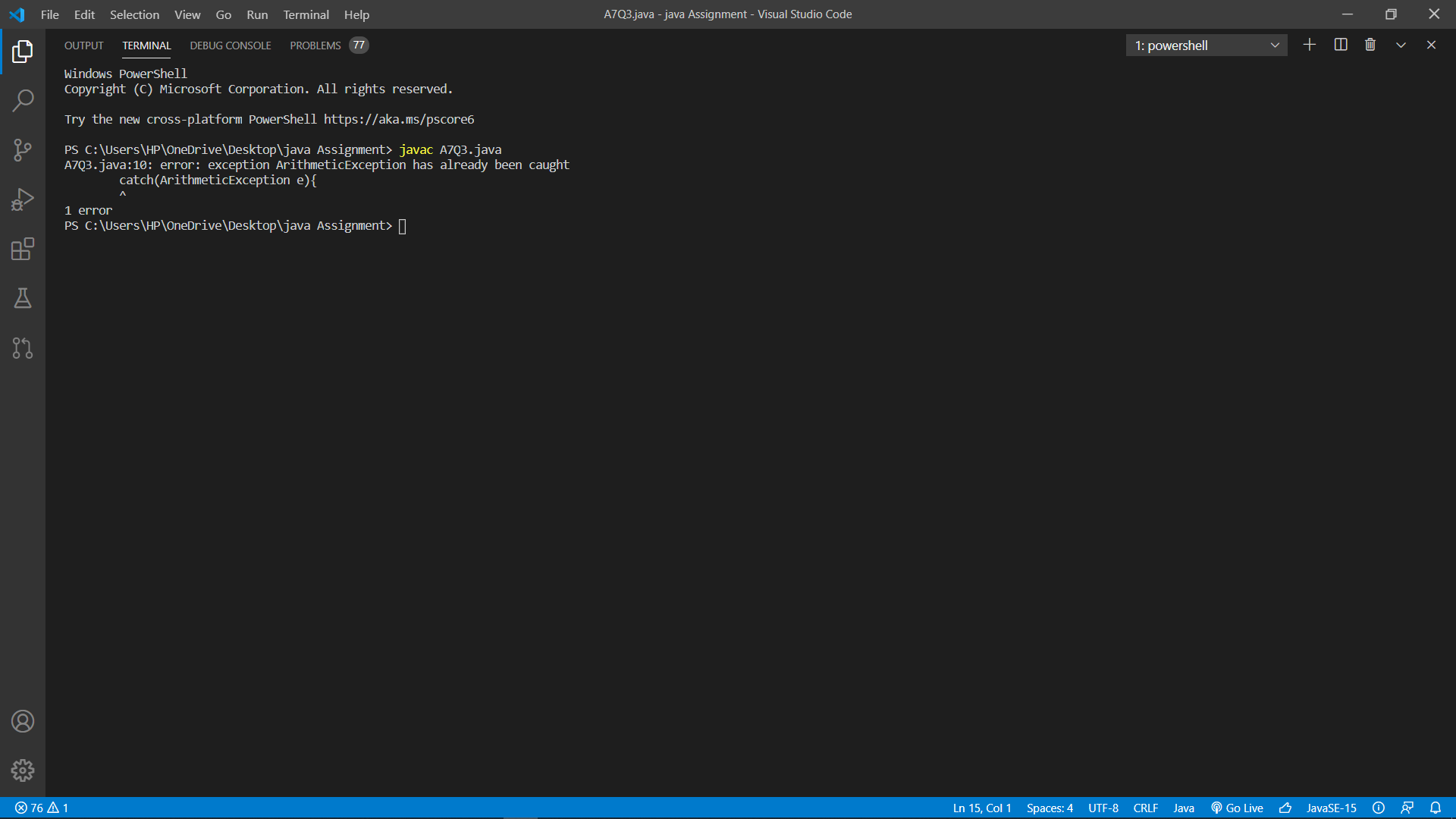
    }

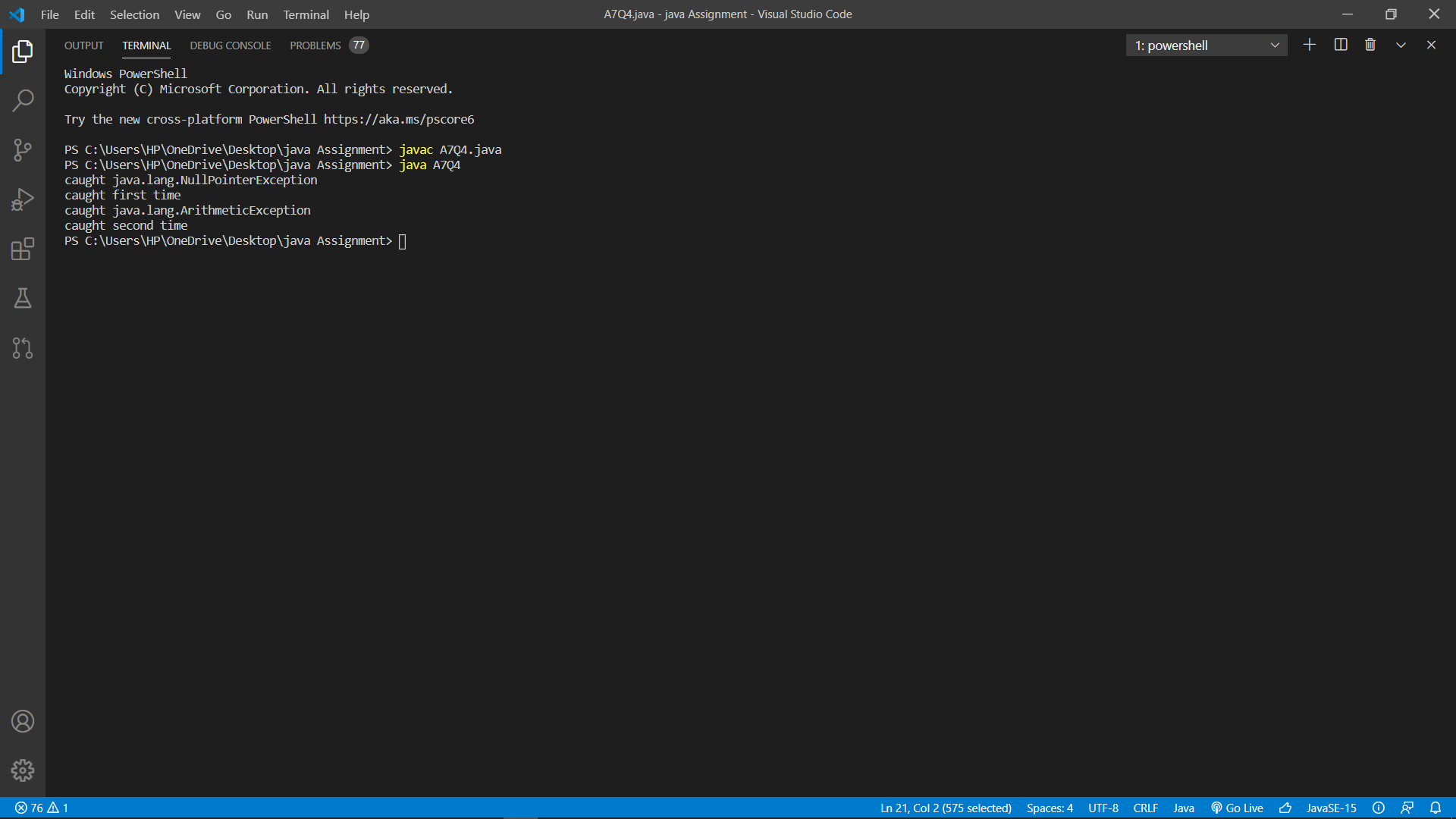
}

Outputs

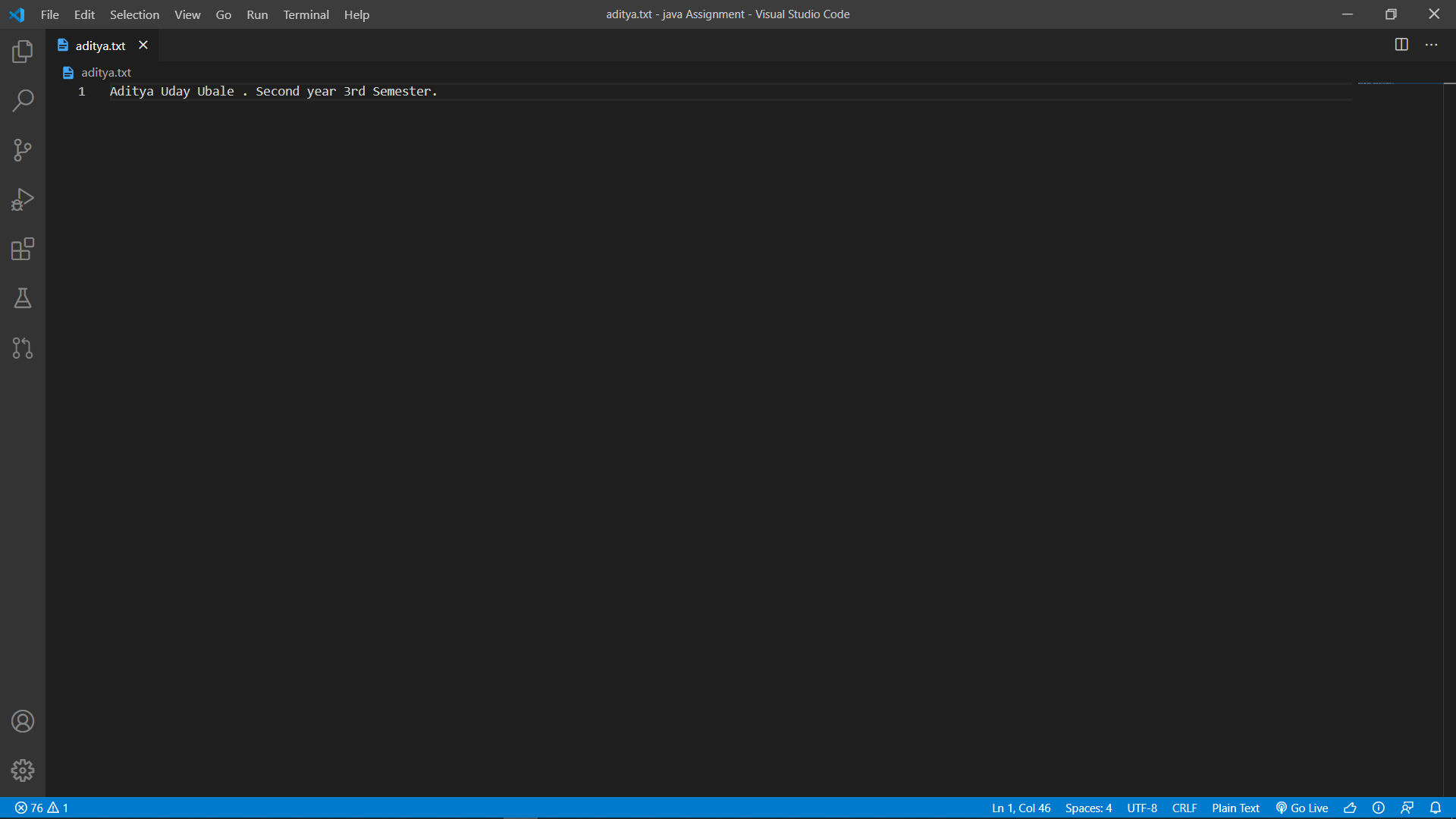
1.) 

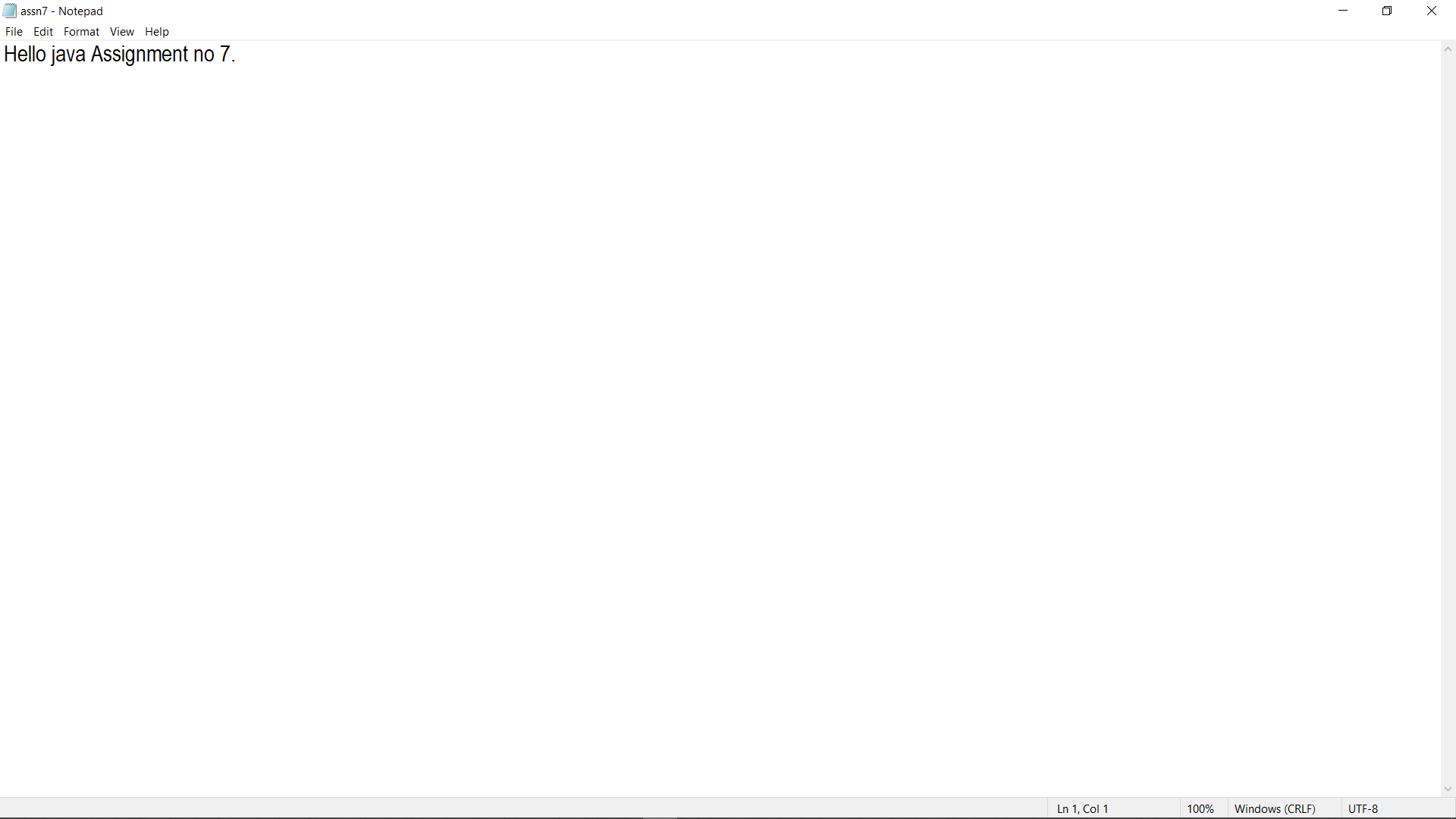
2.) 

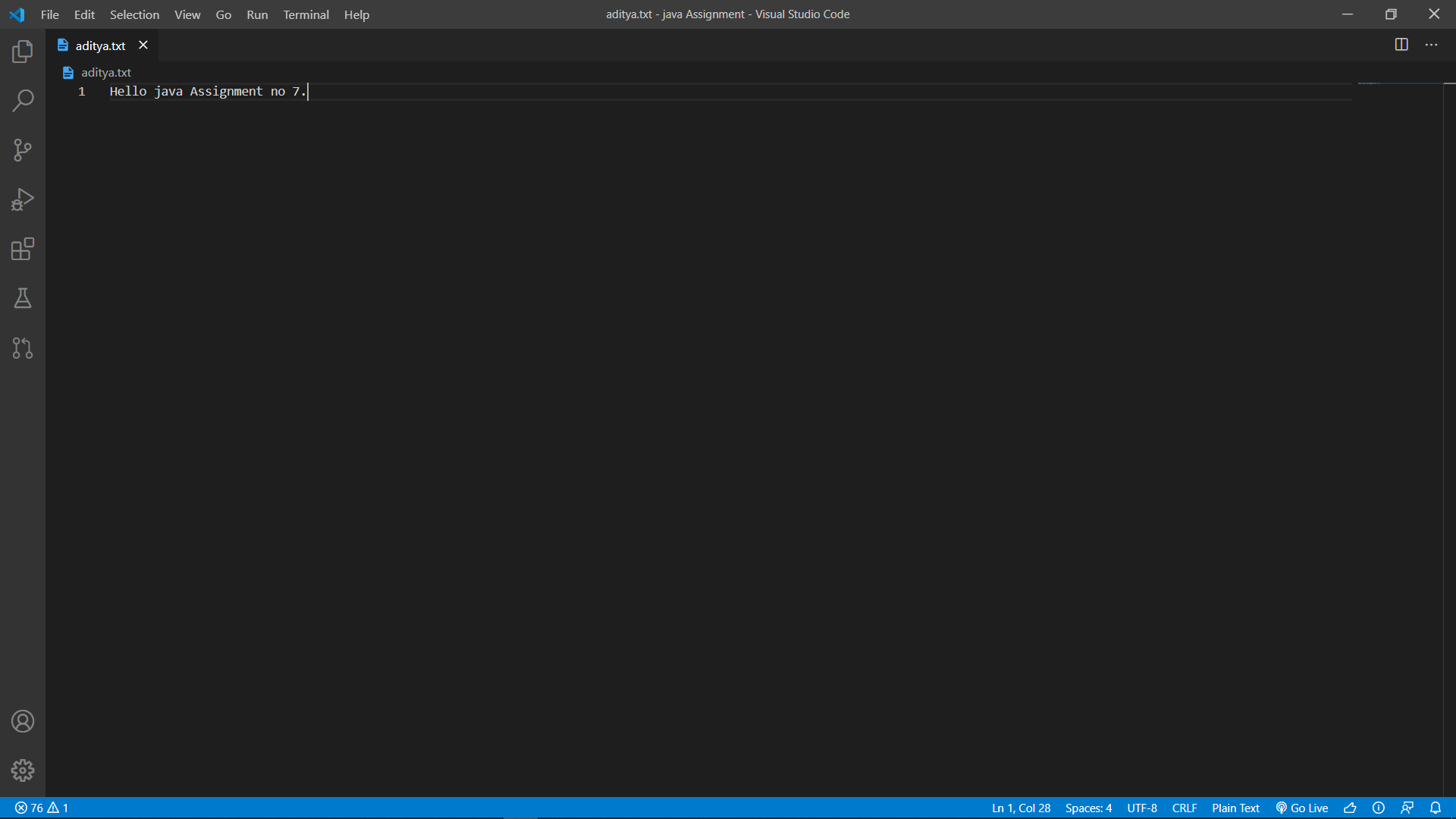
3.) 

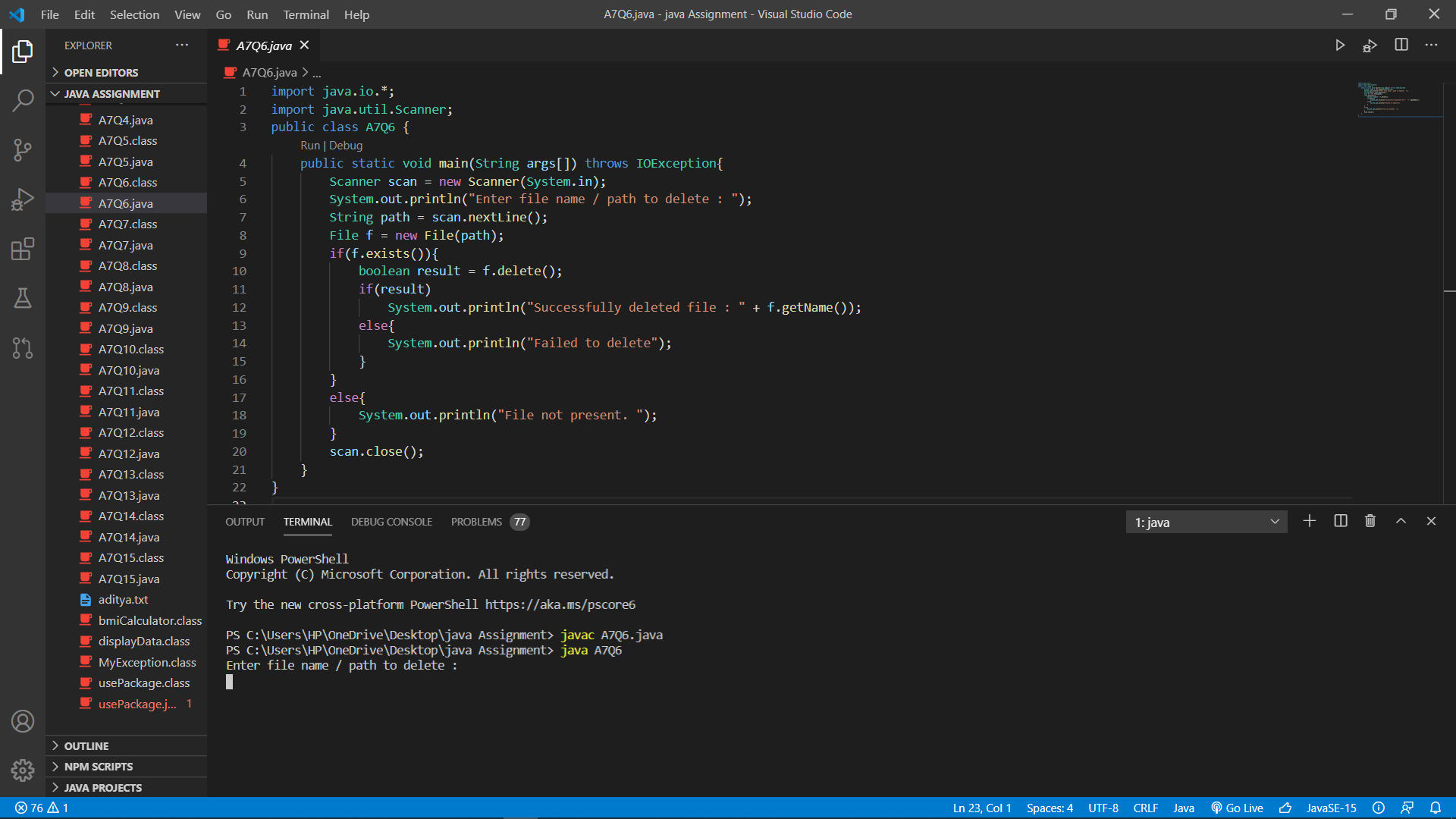
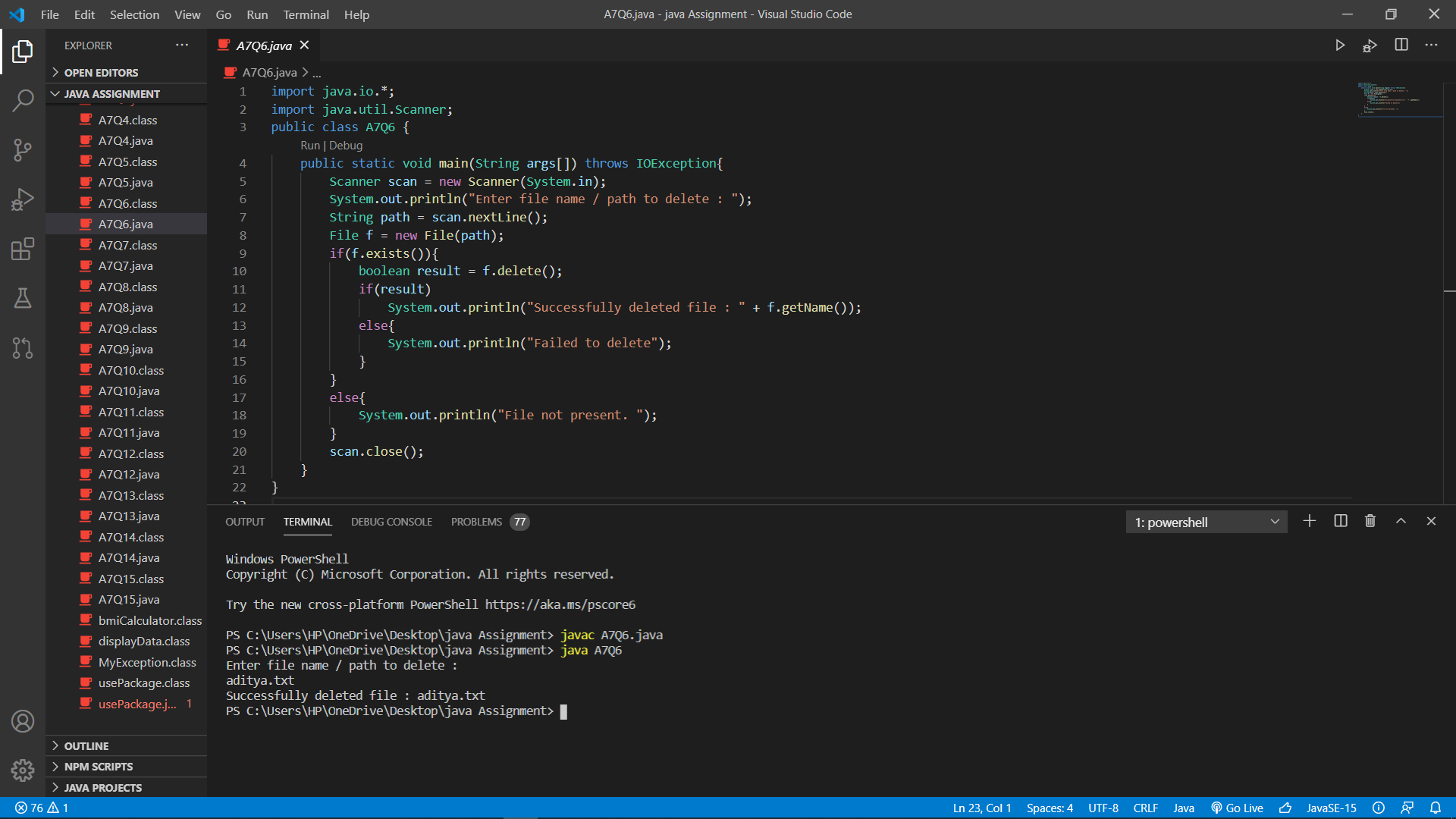
4.) 

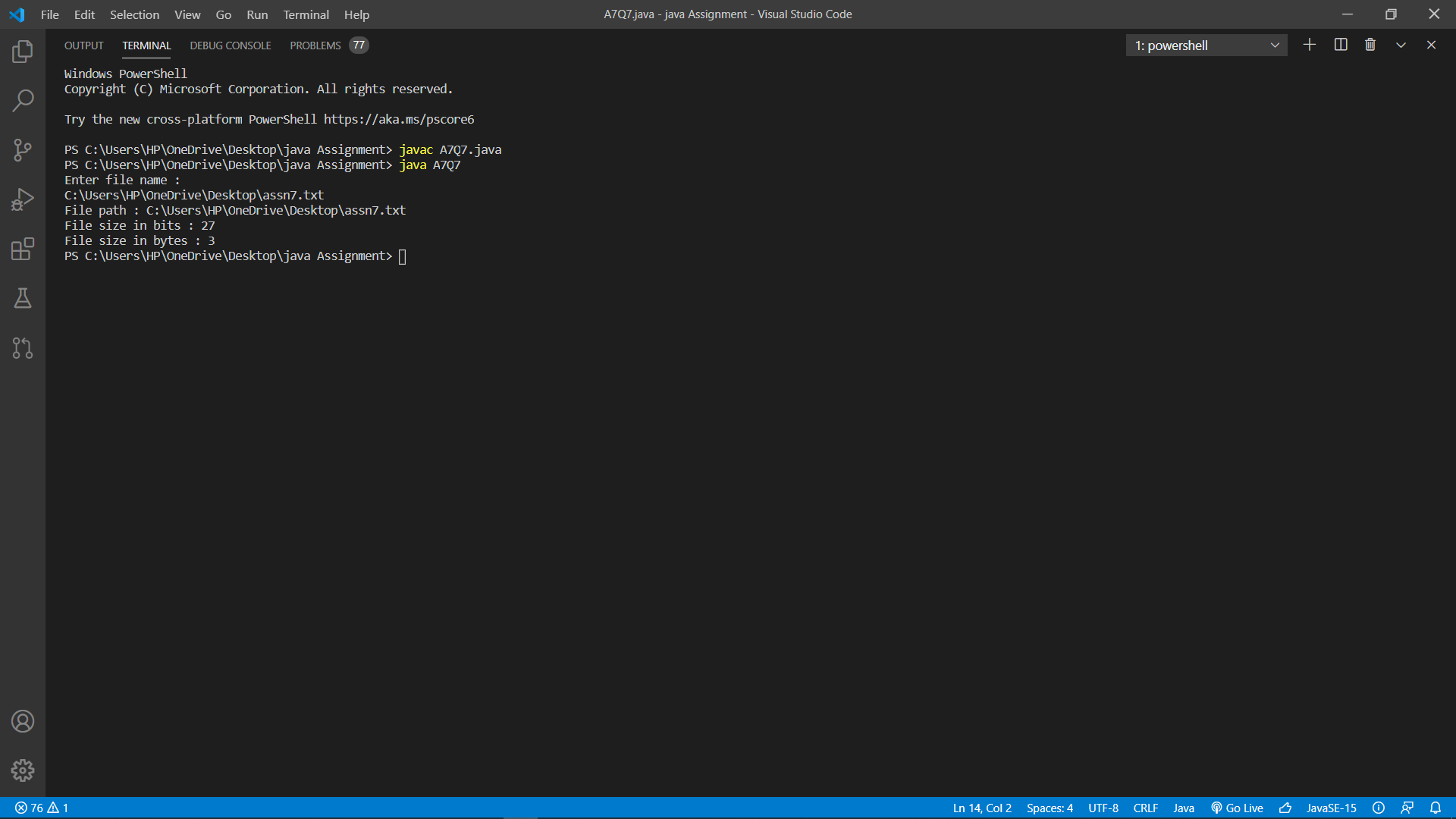
5.)

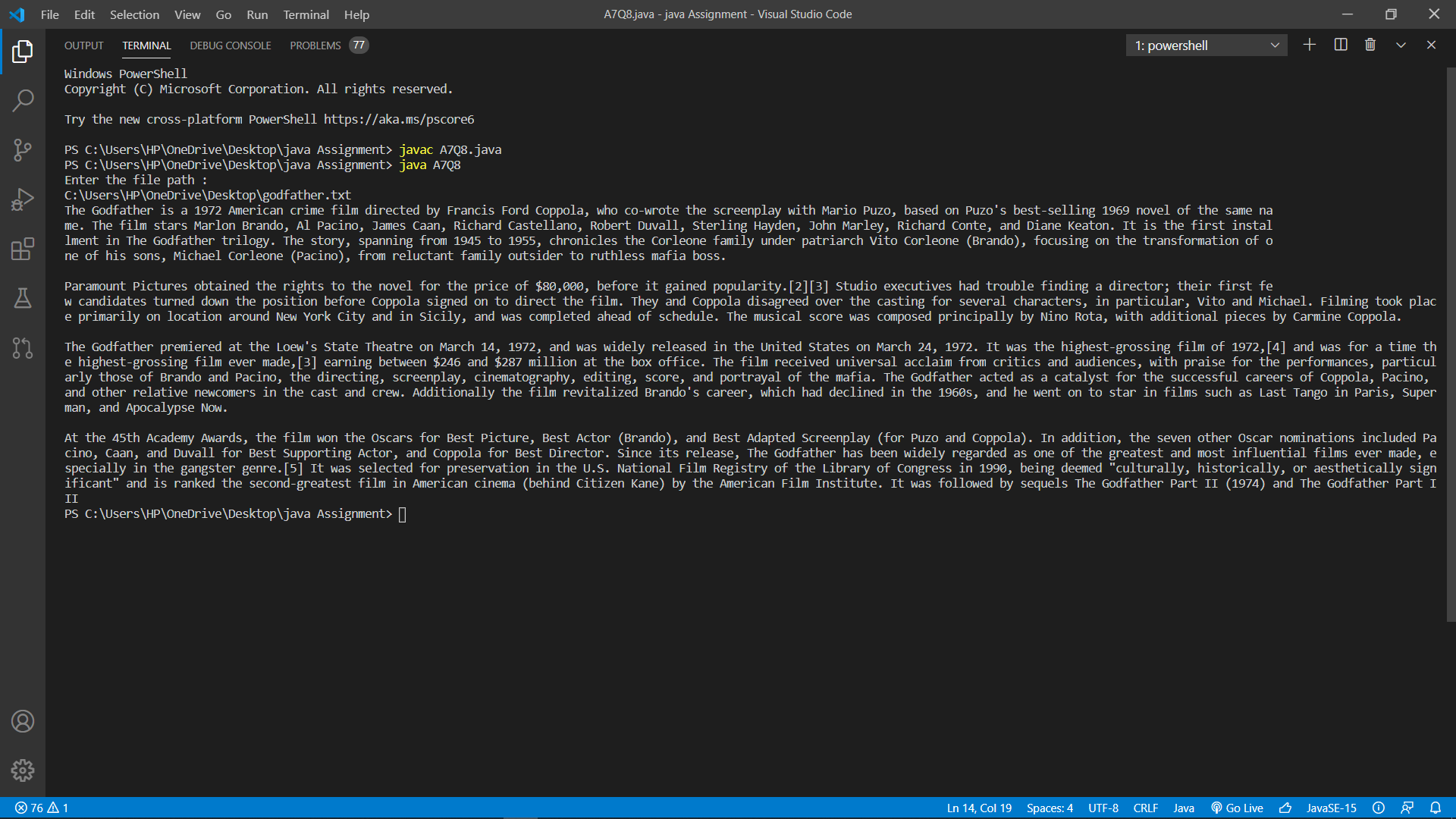


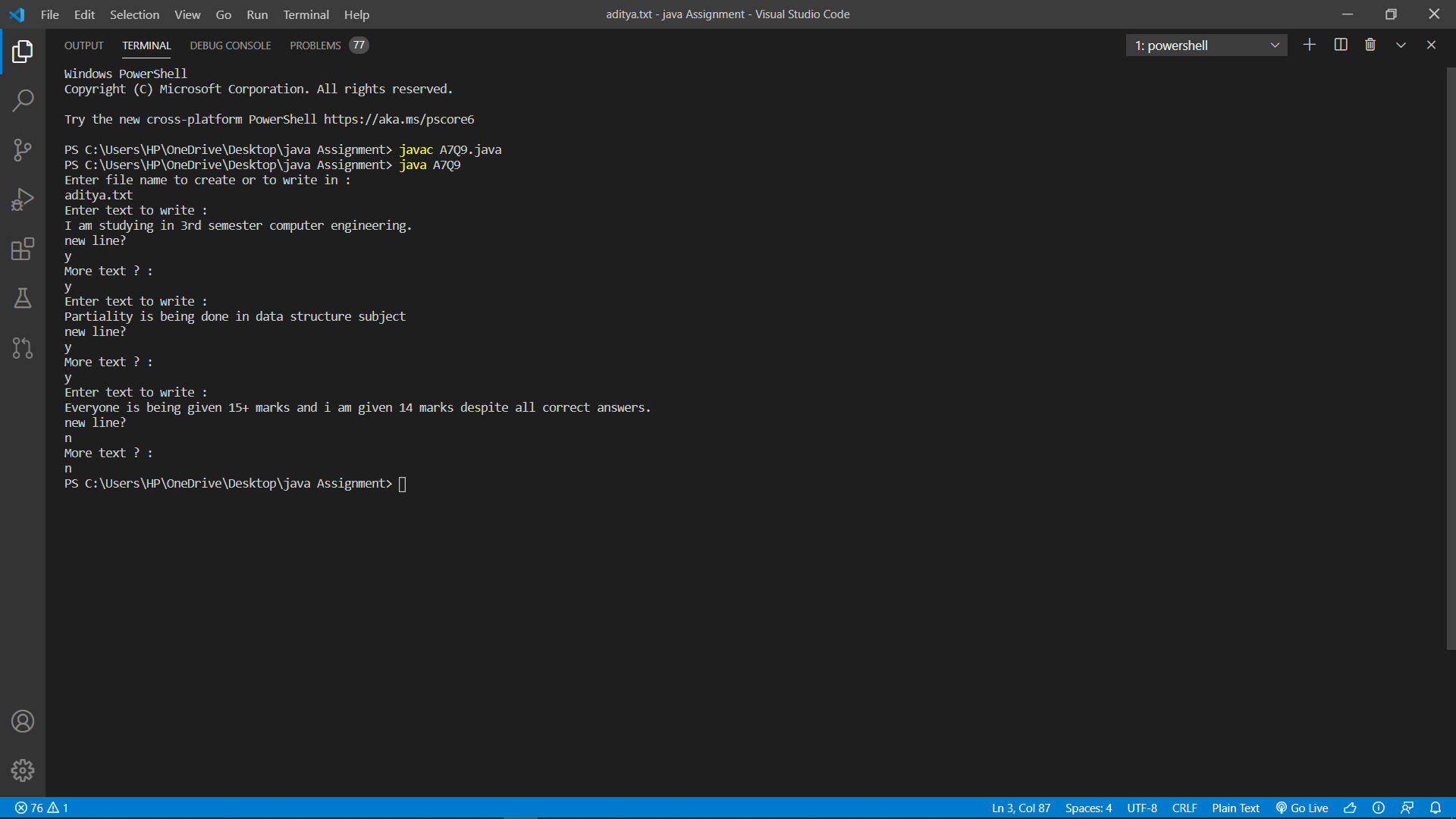


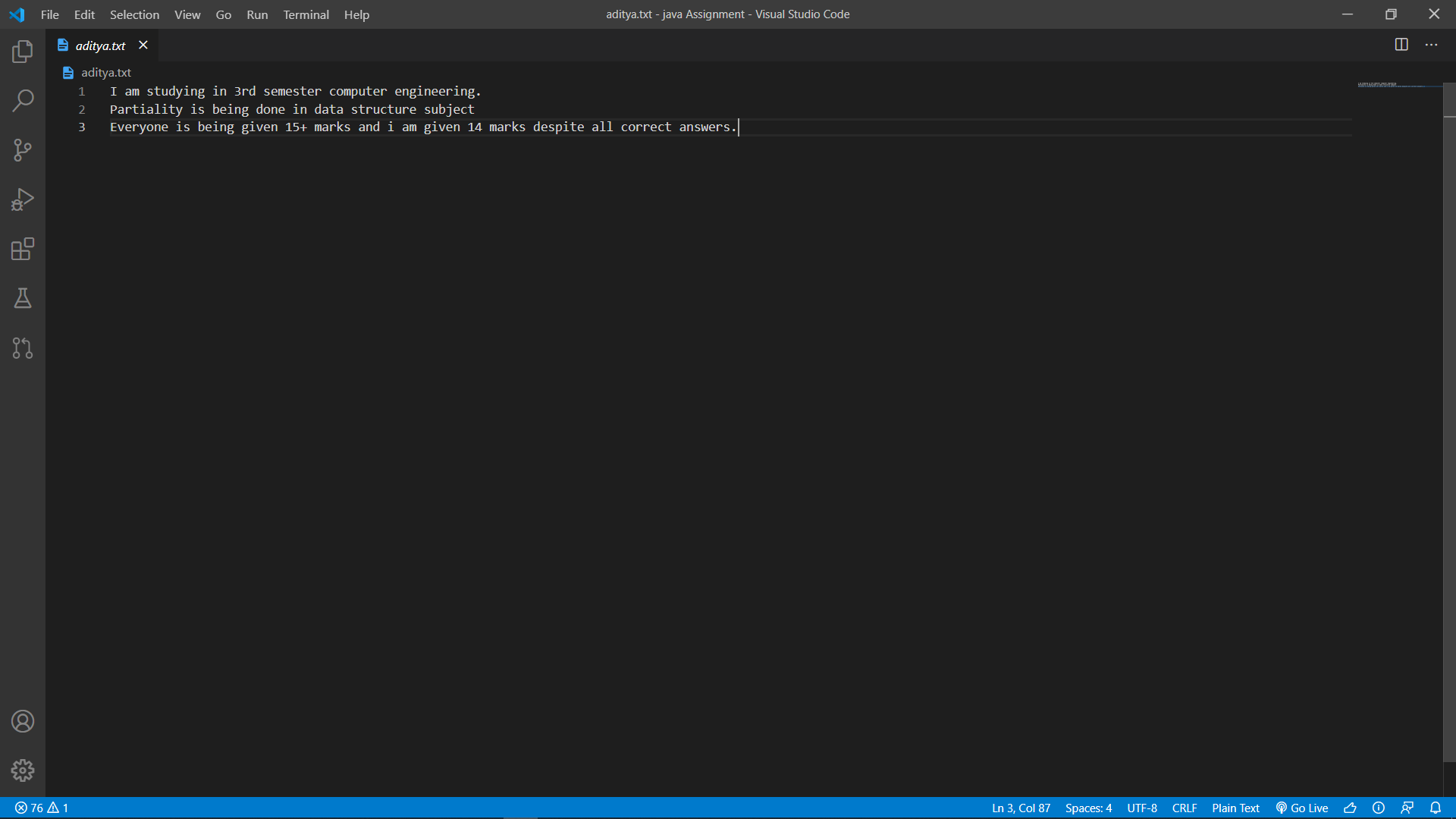


6.) 

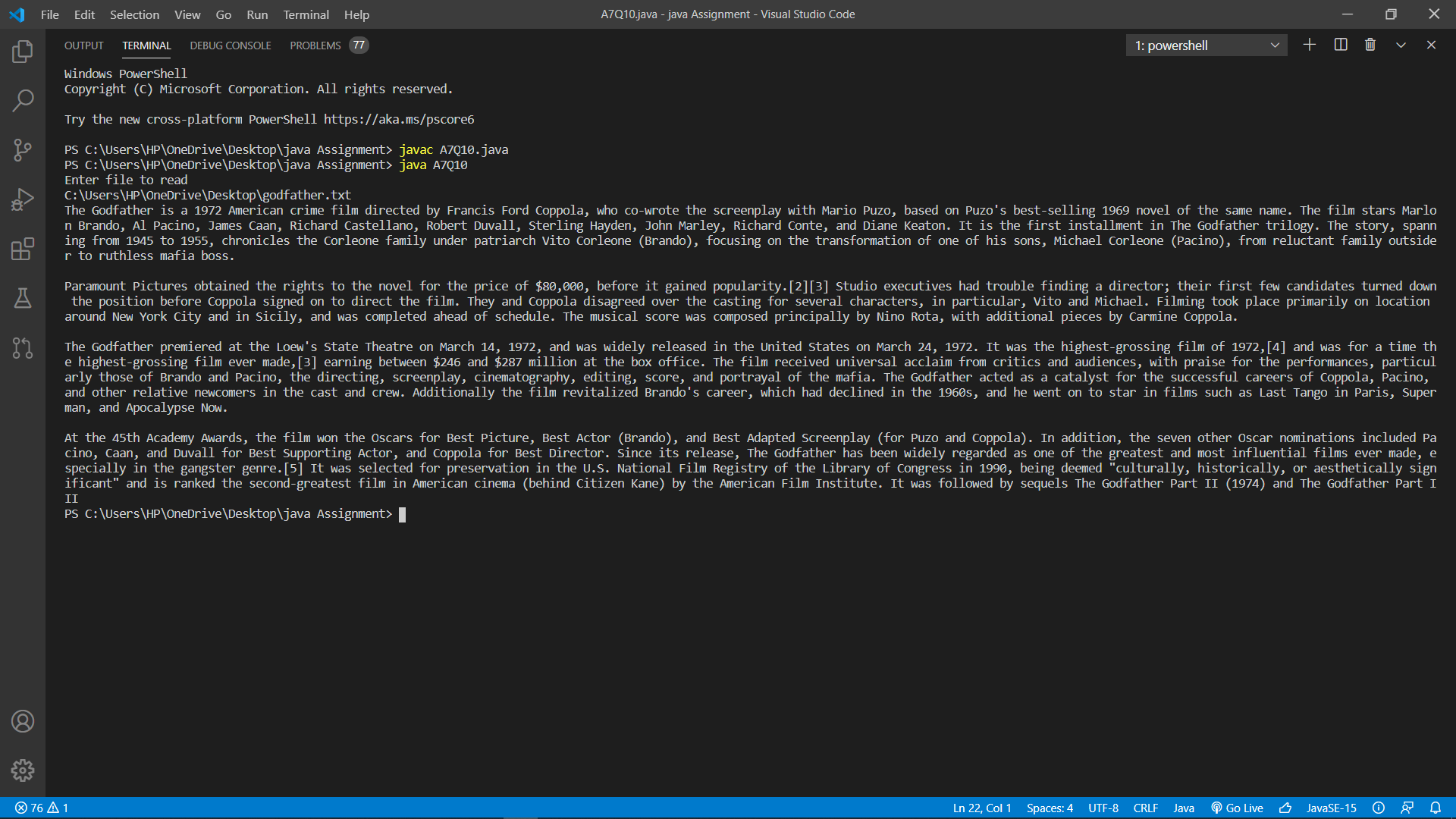
7.) 

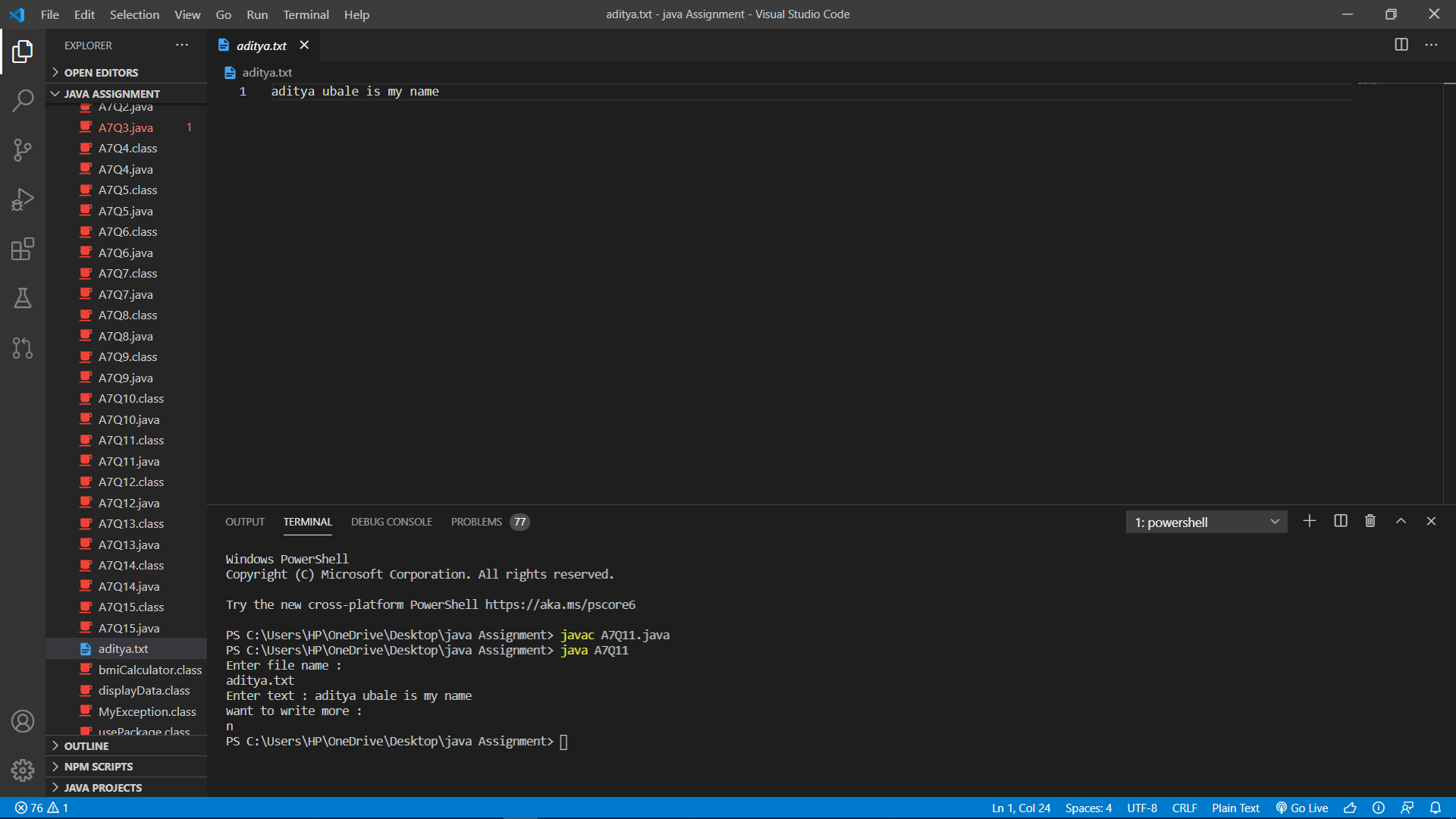
8.) 

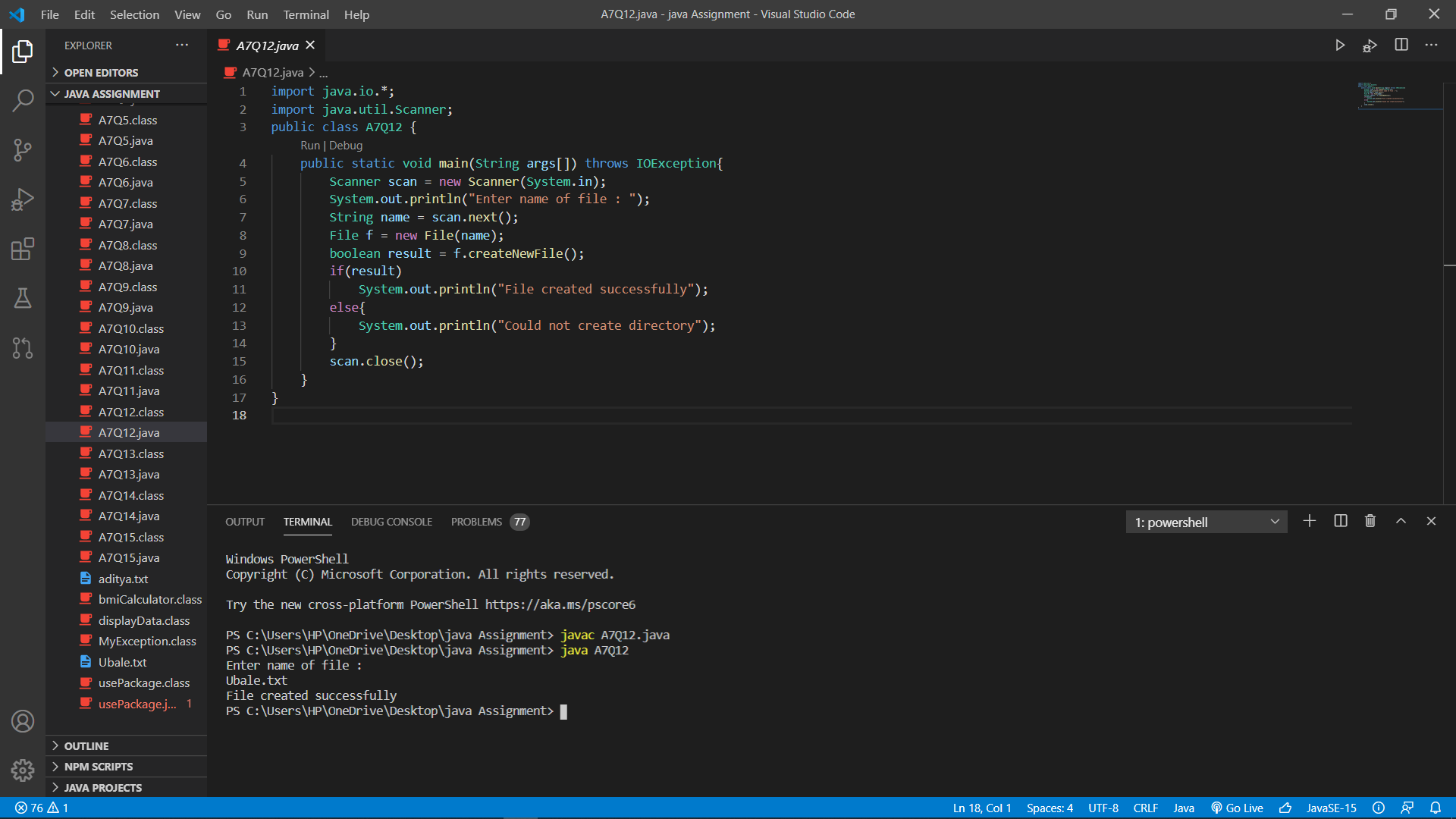
9.) 

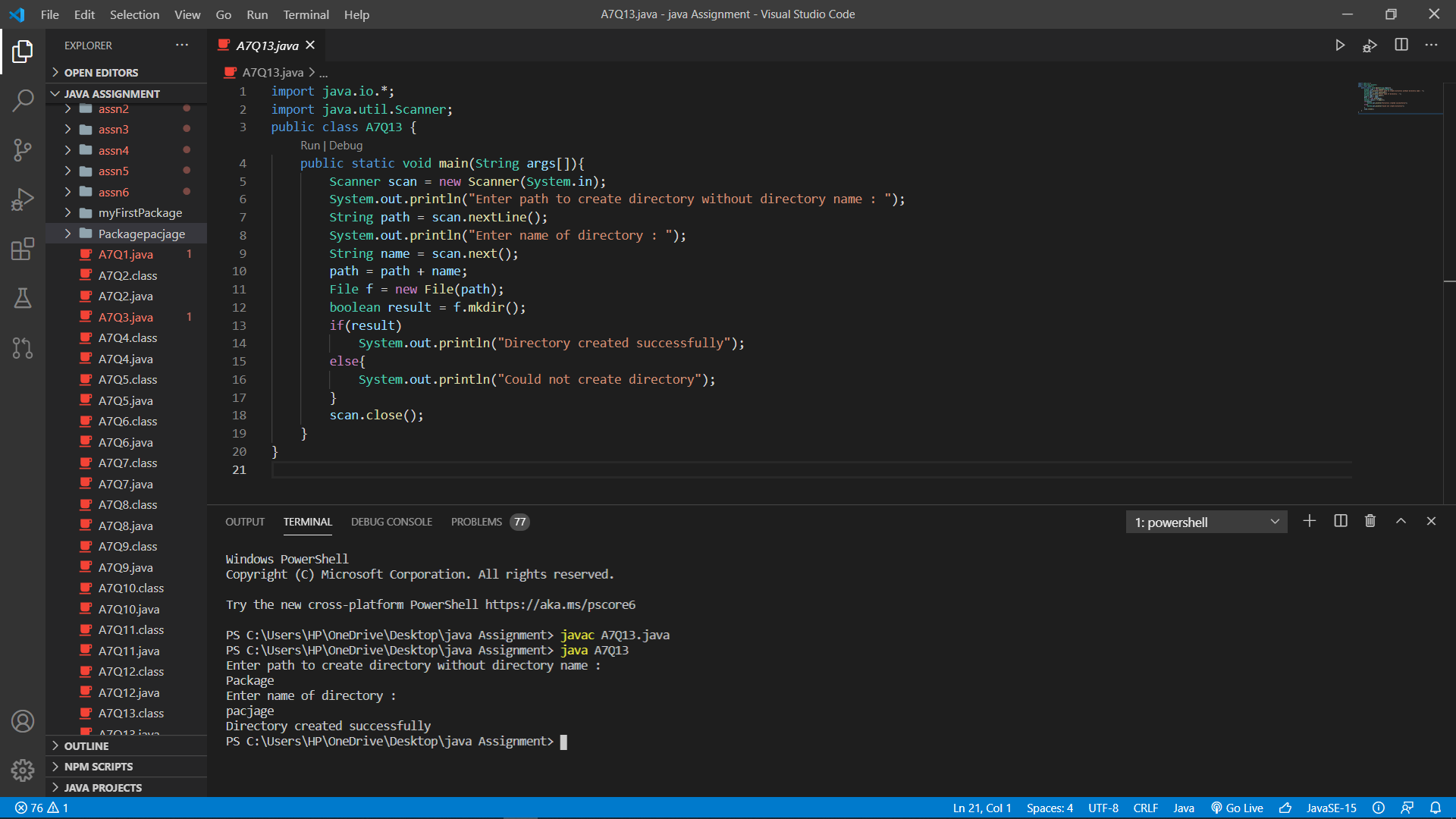


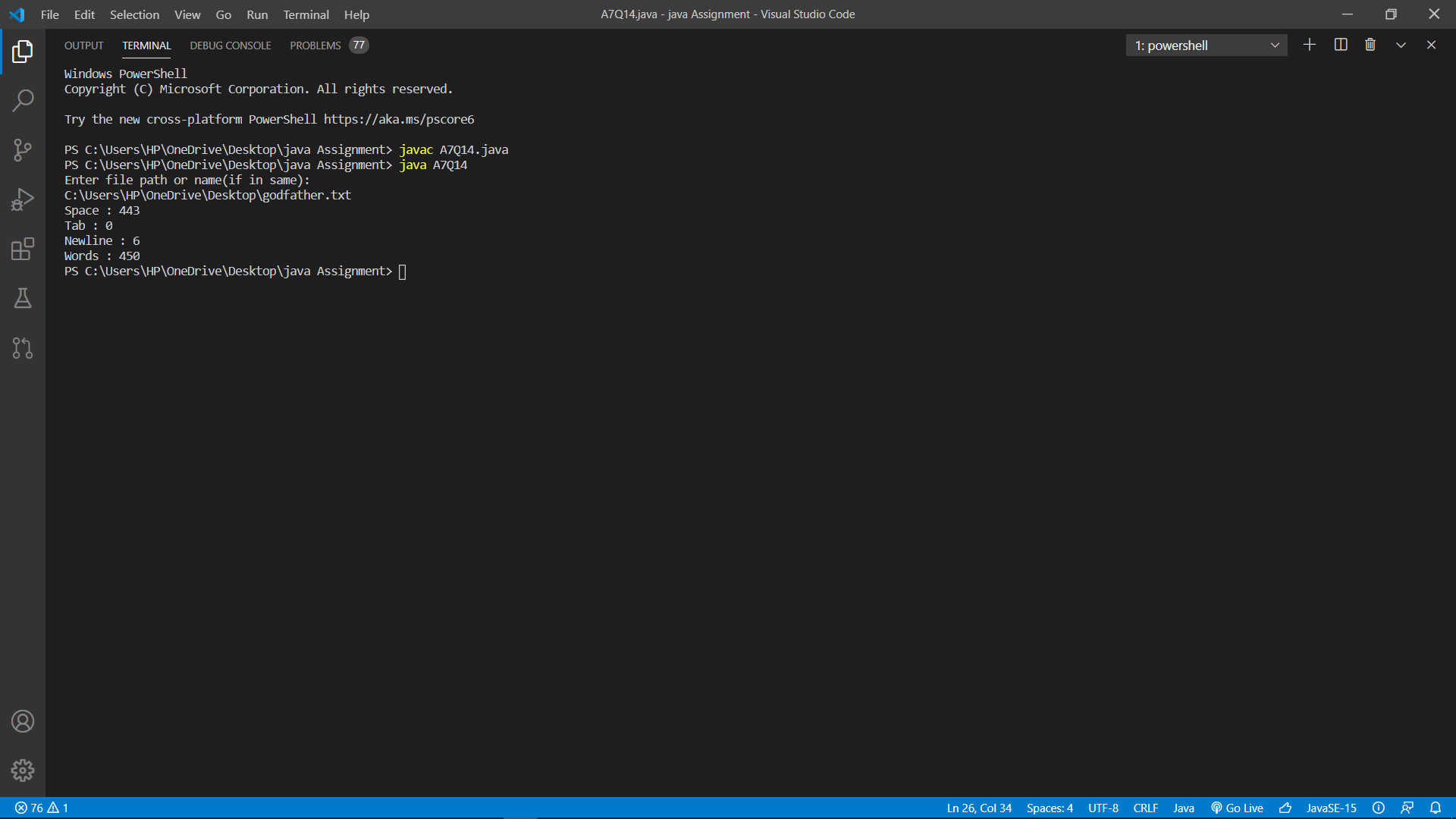
10.)



11.) 

12.) 

13.) 

14.) 

15.) 