Assignment 6

Codes

1.)

class Area{

    double area(double ... *dimensions*){

        int arguments = dimensions.length;

        switch (arguments){

            case 1:

                return Math.PI\*dimensions[0]\*dimensions[0];

            case 2:

                double are=1.0;

                for(double a:dimensions){

                    are\*=a;

                }

                return are;

            default :

                return 0.0;

        }

    }

}

class Volume extends Area{

    double volume(double ... *dimensions*){

        int arguments = dimensions.length;

        switch (arguments){

            case 2:

                return dimensions[1]\*area(dimensions[0]);

            case 3:

                return dimensions[2]\*area(dimensions[0], dimensions[1]);

            default :

                return 0.0;

        }

    }

}

public class A6Q1{

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

        Volume v = new Volume();

        System.out.println("Using volume property by volume class");

        System.out.println("");

        System.out.println("volume of cylinder with r=2 and h=3 :  "+v.volume(2,3));

        System.out.println("");

        System.out.println("volume of cube edge=6 : "+v.volume(6,6,6));

        System.out.println("");

        System.out.println("volume of cuboid with l=3 b=4 h=5 : "+v.volume(3,4,5));

        System.out.println("");

        System.out.println("Using area property by volume class");

        System.out.println("");

        System.out.println("Area of circle with r=5 : "+v.area(5));

        System.out.println("");

        System.out.println("Area of quadilateral with l=5 b=6 : "+v.area(5,6));

    }

}

2.)

class parent{

    int x;

    String name;

    parent(){

        x=10;

        name="Parent";

    }

    void show(){

        System.out.println("Parent class show!");

    }

    void parentMethod(){

        System.out.println("My classname is : "+this.name+" my value is : "+this.x);

    }

}

class child extends parent{

    int y;

    String surname;

    child(){

        y=20;

        surname="Child";

    }

    void show(){

        System.out.println("Child class !how");

    }

    void childMethod(){

        System.out.println("My classname is : "+this.surname+" my value is : "+this.y);

        System.out.println("Calling parent class method using super ");

        super.parentMethod();

    }

}

public class A6Q2{

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

        child c = new child();

        c.show();

        c.childMethod();

    }

}

3.)

class School{

    String name = "Joy Senior Secondary School";

    void show(){

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

    }

}

class Level extends School{

    String name = "Twelfth";

    void show(){

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

        super.show();

    }

}

class Section extends Level{

    String name = "M1";

    void show(){

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

        super.show();

    }

}

class Student extends Section{

    String name = "Aditya";

    void show(){

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

        super.show();

    }

}

public class A6Q3 {

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

        Student s = new Student();

        s.show();

    }

}

4.)

class Shapes{

    double getArea(String *shape*, double ... *dimensions*){

        switch (*shape*){

            case "square":

                return *dimensions*[0]\**dimensions*[0];

            case "circle":

                return Math.PI\**dimensions*[0]\**dimensions*[0];

            case "cube":

                return 6\**dimensions*[0]\**dimensions*[0]\**dimensions*[0];

            case "sphere":

                return 4\*Math.PI\**dimensions*[0]\**dimensions*[0];

            default :

                return 1.0;

        }

    }

    double getVolume(String *shape*, double ... *dimensions*){

        switch (*shape*){

            case "square":

                return 0.0;

            case "circle":

                return 0.0;

            case "cube":

                return *dimensions*[0]\**dimensions*[0]\**dimensions*[0]\*1.0;

            case "sphere":

                return 4\*Math.PI\**dimensions*[0]\**dimensions*[0]\**dimensions*[0]/3;

            default :

                return 1.0;

        }

    }

}

class calculate extends Shapes{

    void area(){

        int x=1;

        while(x<=4){

            switch (x){

                case 1:

                    System.out.println("Area of square : "+getArea("square", 5));

                    ++x;

                    break;

                case 2:

                    System.out.println("Area of circle : "+getArea("circle", 5));

                    ++x;

                    break;

                case 3:

                    System.out.println("Area of cube : "+getArea("cube", 5));

                    ++x;

                    break;

                case 4:

                    System.out.println("Area of sphere : "+getArea("sphere", 5));

                    ++x;

                    break;

            }

        }

    }

    void volume(){

        int x=1;

        while(x<=4){

            switch (x){

                case 1:

                    System.out.println("Volume of square : "+getVolume("square", 5));

                    ++x;

                    break;

                case 2:

                    System.out.println("Volume of circle : "+getVolume("circle", 5));

                    ++x;

                    break;

                case 3:

                    System.out.println("Volume of cube : "+getVolume("cube", 5));

                    ++x;

                    break;

                case 4:

                    System.out.println("Volume of sphere : "+getVolume("sphere", 5));

                    ++x;

                    break;

            }

        }

    }

}

public class A6Q4 {

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

        calculate a = new calculate();

        a.area();

        a.volume();

    }

}

5.)

class Parent{

    void show(){

        System.out.println("Parent class show!");

    }

    void parentMethod(String *name*){

        System.out.println("My classname is : "+*name*);

    }

    void parentMethod(String *name*, int *value*){

        System.out.println("My classname is : "+*name*+" and value is "+*value*);

    }

}

class Child extends Parent{

    void show(){

        System.out.println("Child class show");

    }

    void childMethod(String *name*){

        System.out.println("My classname is : "+*name*);

    }

    void childMethod(String *name*, int *value*){

        System.out.println("My classname is : "+*name*+" and value is "+*value*);

    }

}

public class A6Q5{

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

        Parent p = new Parent();

        System.out.println("Overloading in parent :");

        p.parentMethod("Parent");

        p.parentMethod("Parent", 10);

        Parent q = new Child();

        Child r = new Child();

        System.out.println("Overloading in child :");

        r.childMethod("Child");

        r.childMethod("Child", 20);

        System.out.println("Overriding : ");

        System.out.println("parent ref and parent const");

        p.show();

        System.out.println("parent ref and child const");

        q.show();

        System.out.println("child ref and child const");

        r.show();

    }

}

6.)

import java.util.Scanner;

abstract class Employee{

    abstract void getAmount(double *workTime*, double *rate*);

}

class weeklyEmployee extends Employee{

    void getAmount(double *workTime*, double *rate*){

        double ratepw = *rate*;

        double salary = *workTime*\*ratepw;

        System.out.println("Employee worked for "+*workTime*+" weeks. Salary payable is : "+salary);

    }

}

class hourlyEmployee extends Employee{

    void getAmount(double *workTime*, double *rate*){

        double rateph = *rate*;

        double salary = *workTime*\*rateph;

        System.out.println("Employee worked for "+*workTime*+" hours. Salary payable is : "+salary);

    }

}

public class A6Q6{

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

        double rate , workTime;

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter employee type : ");

        String type = scan.nextLine();

        switch(type){

            case "hourly":

                System.out.println("Enter rate per hour : ");

                rate = scan.nextDouble();

                System.out.println("Enter no of hours worked : ");

                workTime = scan.nextDouble();

                hourlyEmployee h = new hourlyEmployee();

                h.getAmount(workTime, rate);

                break;

            case "weekly":

                System.out.println("Enter rate per week : ");

                rate = scan.nextDouble();

                System.out.println("Enter no of weeks worked : ");

                workTime = scan.nextDouble();

                weeklyEmployee w = new weeklyEmployee();

                w.getAmount(workTime, rate);

                break;

        }

        scan.close();

    }

}

7.)

import java.util.Scanner;

interface payable{

    double getAmount(double *total*);

}

class invoice implements payable{

    public double getAmount(double *total*){

        double amount;

        amount = *total*\*(1+0.11);

        return amount;

    }

    public void show(double *am*){

        System.out.println("Invoice payment : "+getAmount(*am*));

    }

}

class employee implements payable{

    public double getAmount(double *total*){

        double amount;

        amount = *total*\*(0.15+0.03+0.11+1);

        return amount;

    }

    public void show(double *am*){

        System.out.println("Employee salary payable : "+getAmount(*am*));

    }

}

public class A6Q7{

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

        Scanner scan = new Scanner(System.in);

        invoice i = new invoice();

        employee e = new employee();

        System.out.println("Enter invoice total : ");

        double am1= scan.nextDouble();

        System.out.println("Enter employee basic : ");

        double am2 = scan.nextDouble();

        i.show(am1);

        e.show(am2);

        scan.close();

    }

}

8.)

class os{

    void performance(){

        System.out.println("os class show!");

    }

}

class windows extends os{

    void performance(){

        System.out.println("windows class show");

    }

}

public class A6Q8 {

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

        os p = new os();

        os q = new windows();

        windows r = new windows();

        System.out.println("os ref and os const");

        p.performance();

        System.out.println("os ref and windows const");

        q.performance();

        System.out.println("windows ref and windows const");

        r.performance();

    }

}

9.)

interface vehicle{

    String getColor();

    String getNumber();

    double getConsumption();

}

class twoWheeler implements vehicle{

    String color="black";

    String model\_name\_number="Royal Enfield Classic 500";

    double consumption=52.34;

    public String getColor(){

        return this.color;

    }

    public String getNumber(){

        return this.model\_name\_number;

    }

    public double getConsumption(){

        return this.consumption;

    }

}

class fourWheeler implements vehicle{

    String color="black";

    String model\_name\_number="Toyota Fortuner v2.5";

    double consumption=16.00;

    public String getColor(){

        return this.color;

    }

    public String getNumber(){

        return this.model\_name\_number;

    }

    public double getConsumption(){

        return this.consumption;

    }

}

public class A6Q9 {

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

        twoWheeler t = new twoWheeler();

        System.out.println("Color of twowheeler: "+t.getColor());

        System.out.println("Model of twowheeler:: "+t.getNumber());

        System.out.println("Consumption of twowheeler:: "+t.getConsumption());

        fourWheeler f = new fourWheeler();

        System.out.println("Color of fourwheeler: "+f.getColor());

        System.out.println("Model of fourwheeler:: "+f.getNumber());

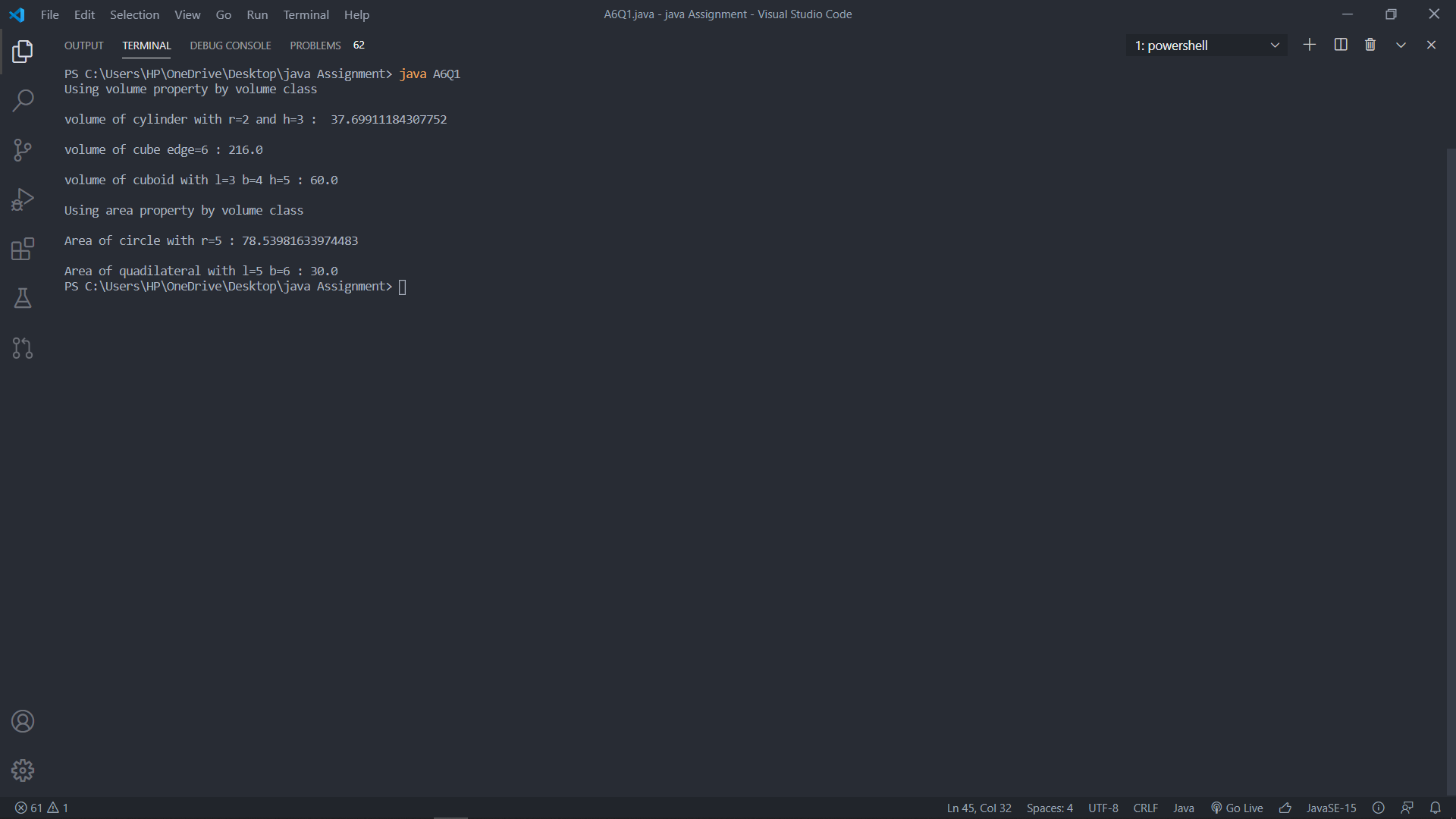
        System.out.println("Consumption of fourwheeler:: "+f.getConsumption());

    }

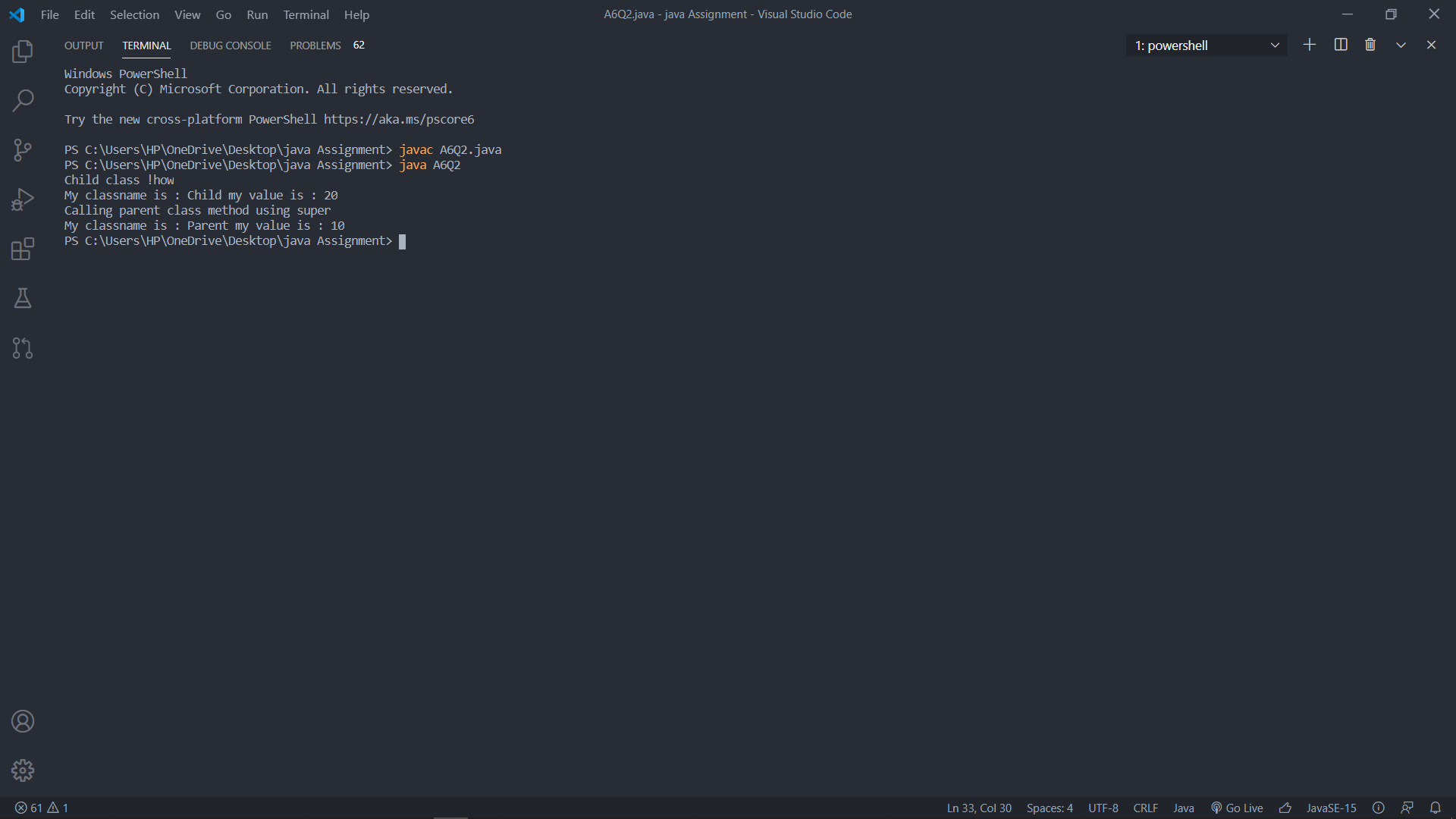
}

Outputs

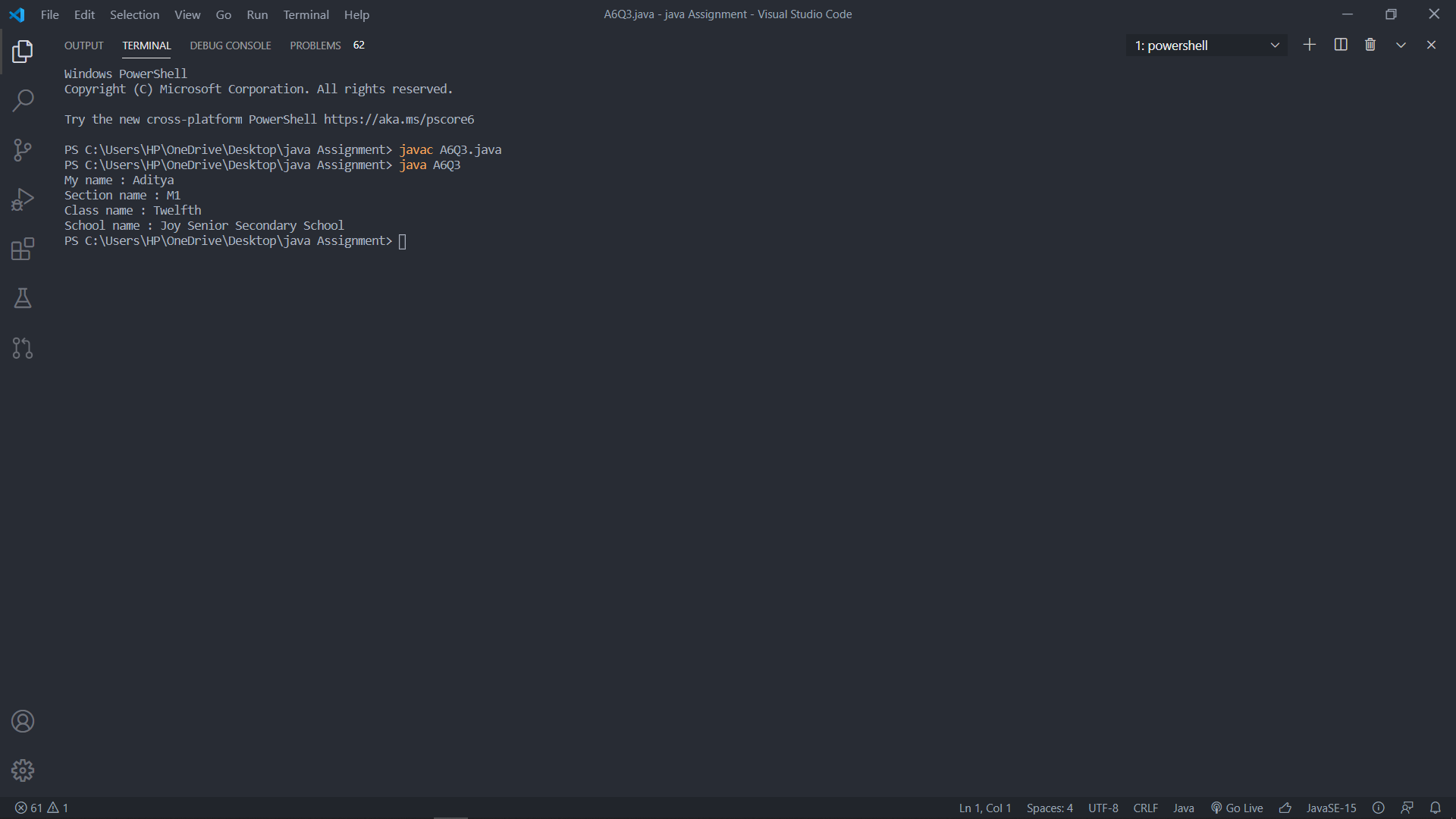
1.)



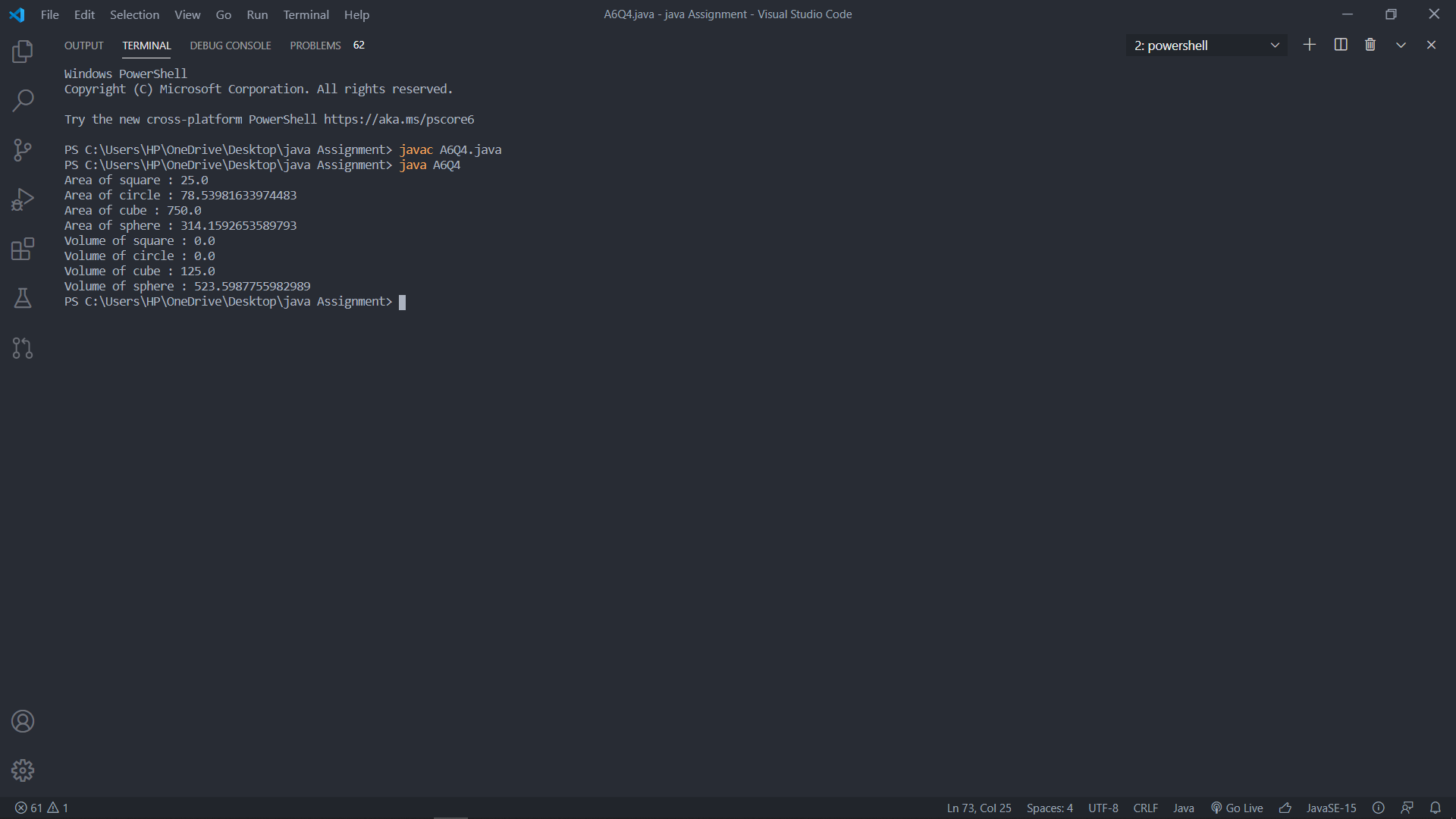
2.)



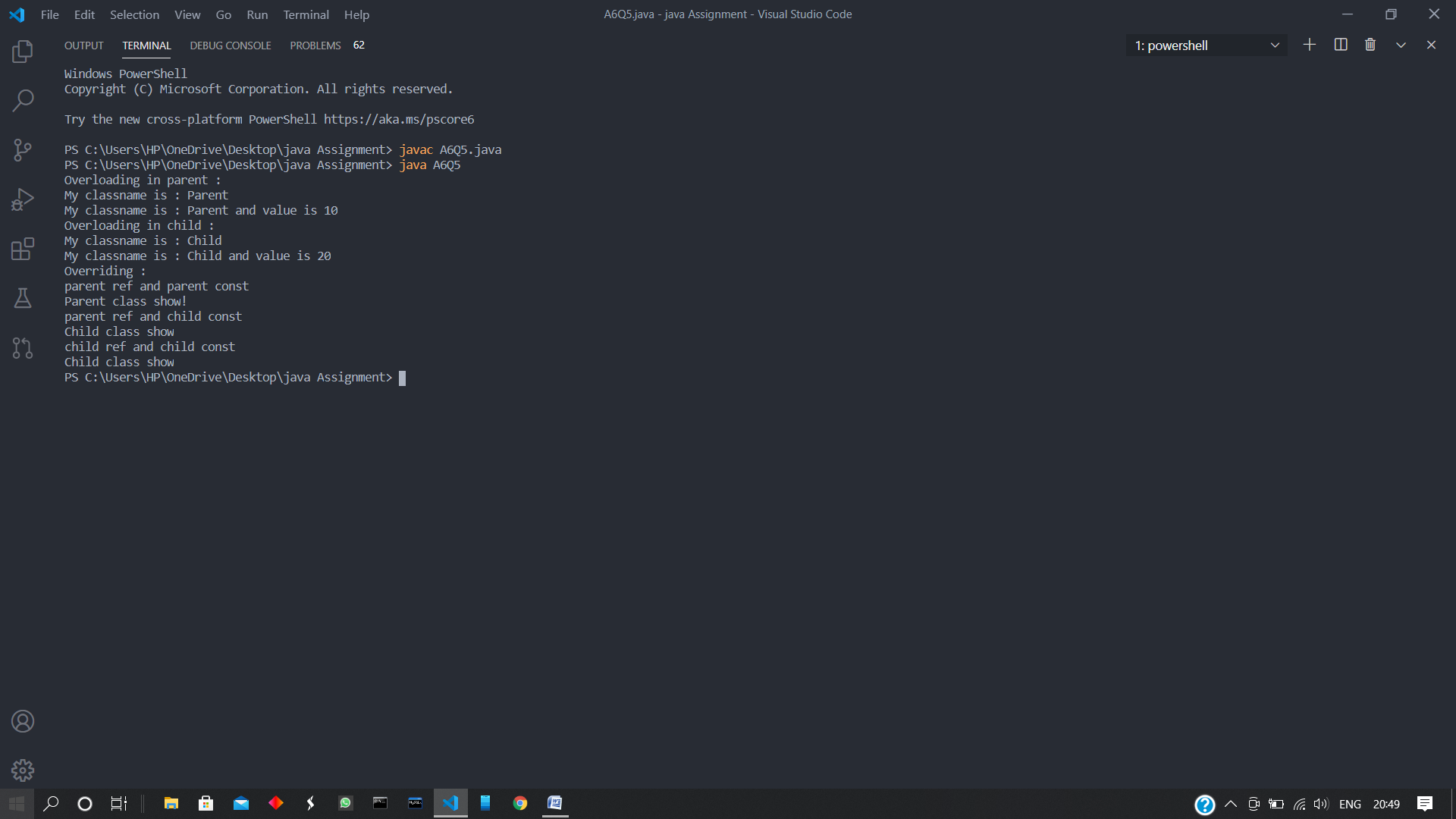
3.)



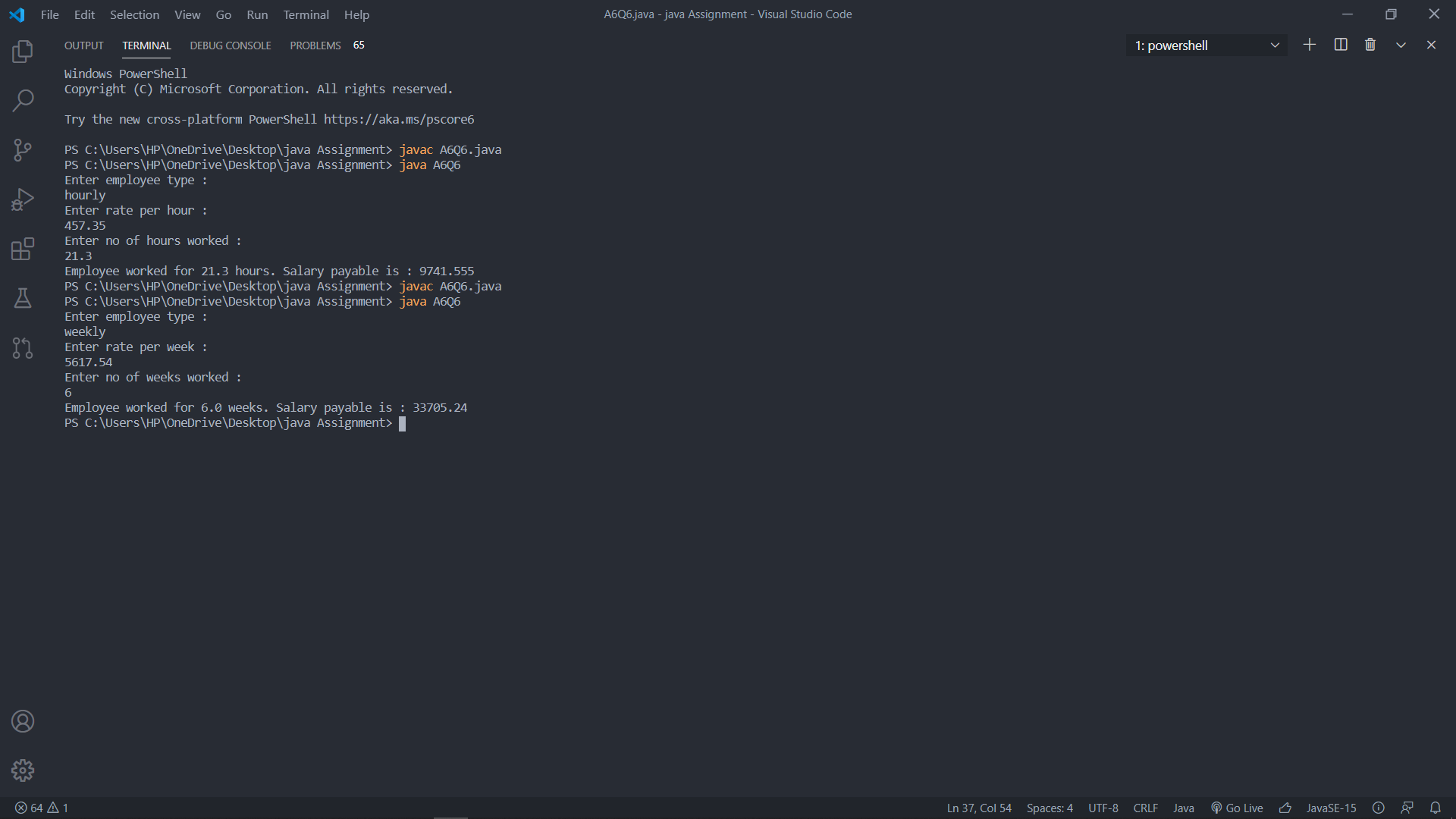
4.)



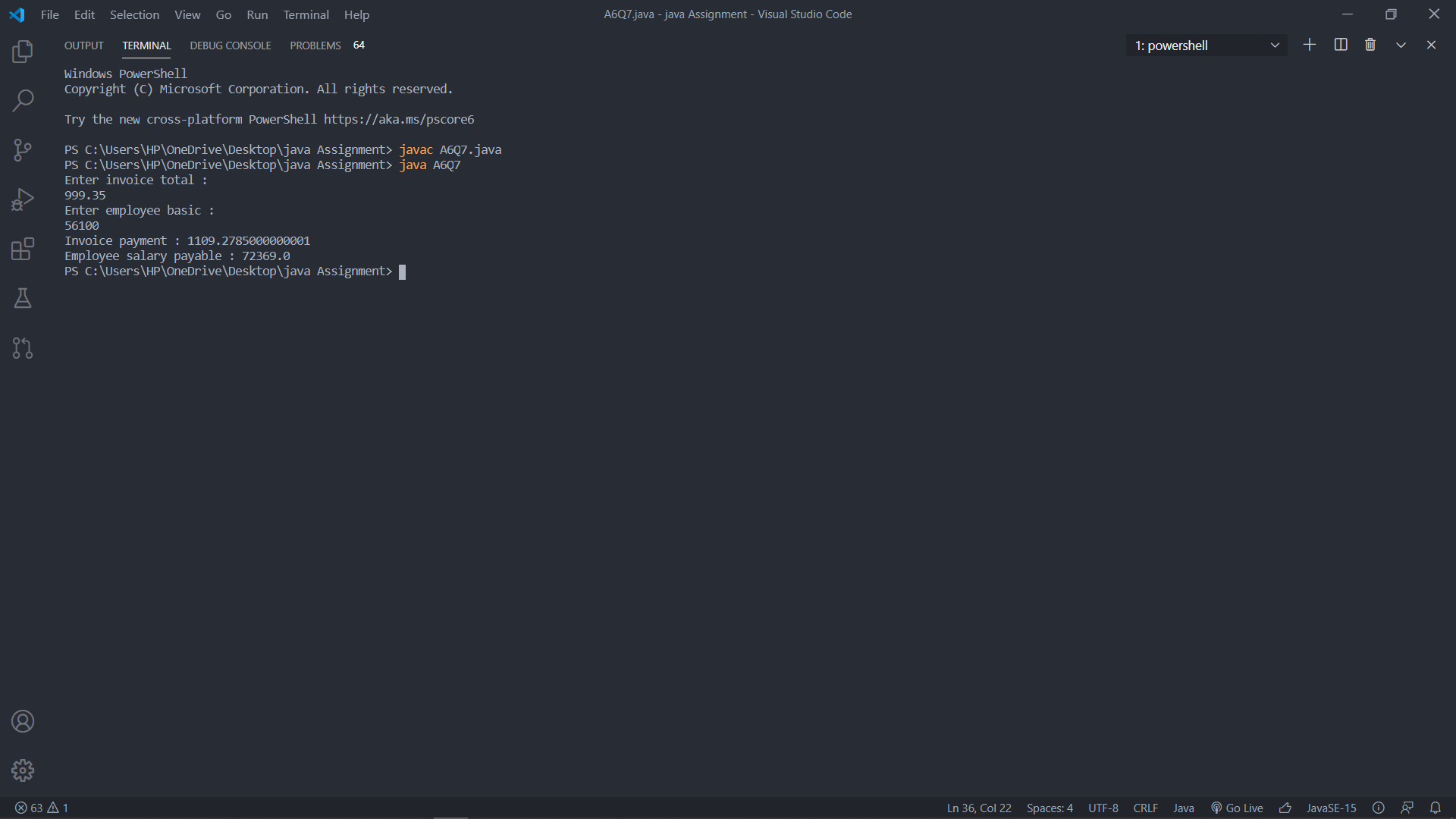
5.)



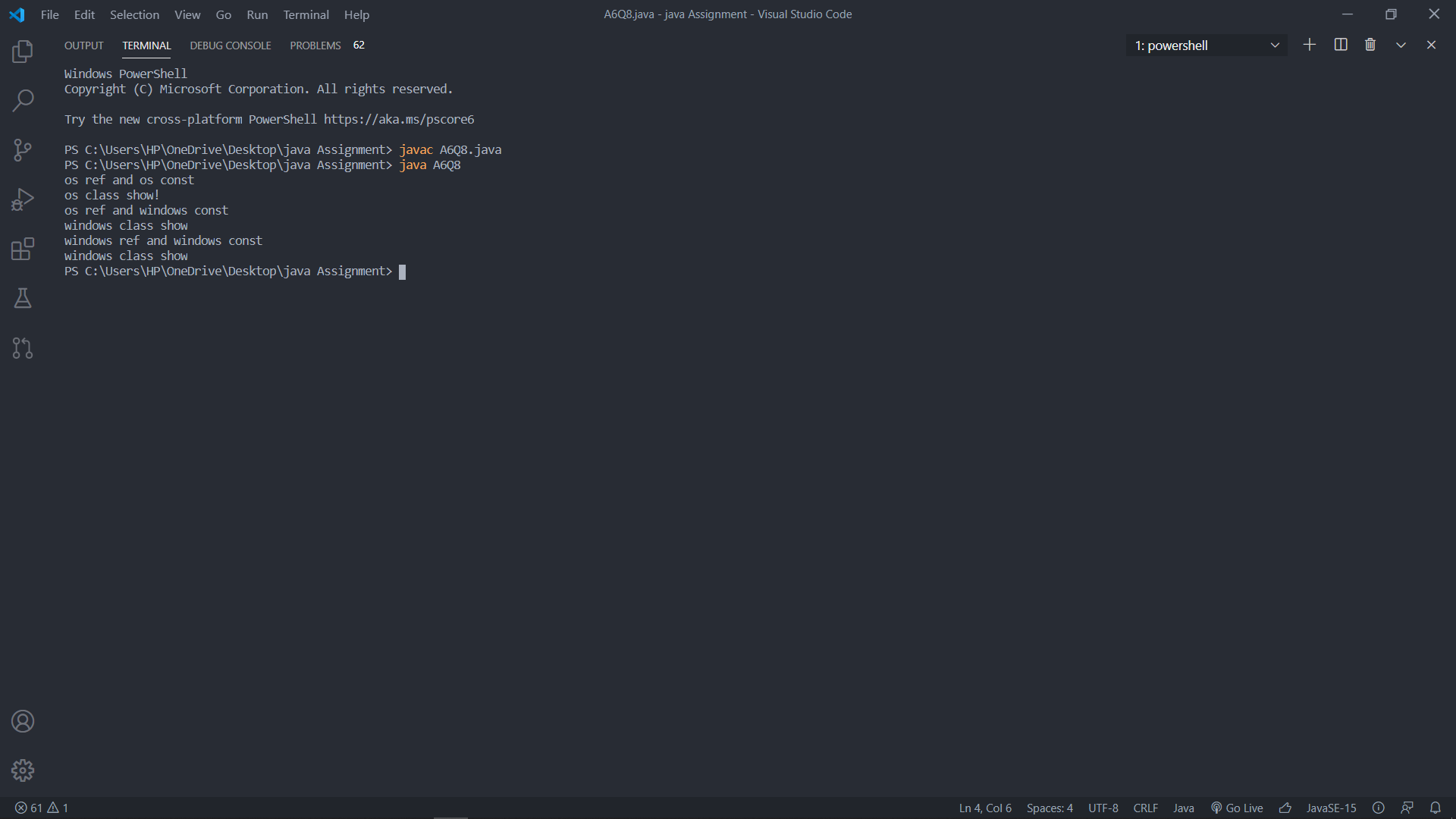
6.)



7.)



8.)



9.)

