Assignment – 1 & 2

Name : Rana Pratipalsinh Hardevsinh

Roll No. : 3161

Division : B

Subject : Java Practical

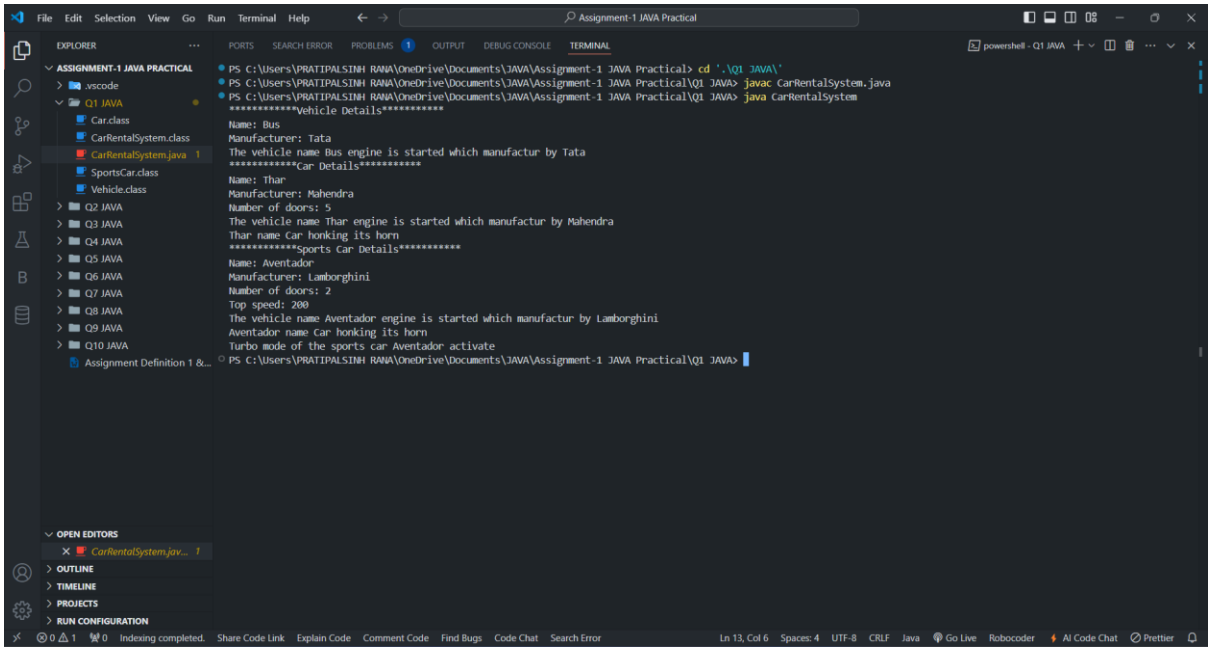
Q.1:

```
class Vehicle{

   String name;
   String manufacturer;

   Vehicle(String name, String manufacturer){
      this.name = name;
      this.manufacturer = manufacturer;
   }

   void startEngine(){
      System.out.println("The vehicle name "+ name +" engine is started which
manufactur by "+ manufacturer);
   }


}

class Car extends Vehicle{

   int numDoors;

   Car(String name, String manufacturer, int numDoors){
      super(name,manufacturer);
      this.numDoors = numDoors;
```

```
    }

    void honkHorn(){
        System.out.println(name + " name Car honking its horn");
    }
}

class SportsCar extends Car{

    int topSpeed;

    SportsCar(String name, String manufacturer, int numDoors,int topSpeed){
        super(name,manufacturer,numDoors);
        this.topSpeed = topSpeed;
    }

    void activateTurbo(){
        System.out.println("Turbo mode of the sports car "+ name +" activate");
    }
}

class CarRentalSystem{
    public static void main(String[] args) {
        Vehicle v1 = new Vehicle("Bus","Tata");
        Car c1 = new Car("Thar","Mahendra",5);
        SportsCar sc1 =  new SportsCar("Aventador","Lamborghini",2,200);
```

```
        System.out.println("**********Vehicle Details**********");
        System.out.println("Name: "+ v1.name);
        System.out.println("Manufacturer: "+ v1.manufacturer);
        v1.startEngine();

        System.out.println("**********Car Details**********");
        System.out.println("Name: "+ c1.name);
        System.out.println("Manufacturer: "+ c1.manufacturer);
        System.out.println("Number of doors: "+ c1.numDoors);
        c1.startEngine();
        c1.honkHorn();

        System.out.println("**********Sports Car Details**********");
        System.out.println("Name: "+ sc1.name);
        System.out.println("Manufacturer: "+ sc1.manufacturer);
        System.out.println("Number of doors: "+ sc1.numDoors);
        System.out.println("Top speed: "+ sc1.topSpeed);
        sc1.startEngine();
        sc1.honkHorn();
        sc1.activateTurbo();
    }
}
```

Output:



Q.2:

```
abstract class Thali {

  private double price;

  Thali() {
     this.price = 0.0;
  }

  abstract void addSabji(double price);

  abstract void addDal(double price);
```

```
    abstract void addRice(double price);

    abstract void addRoti(double price);

    void makeThali() {
        System.out.println("The Veg Thali will be ready in 30 minutes");
    }

    public double getPrice() {
        return this.price;
    }

    protected void setPrice(double price) {
        this.price += price;
    }
}

class GujaratiThali extends Thali {

    public void addSabji(double price) {
        System.out.println("Sabji : "+price);
        setPrice(price);
    }

    public void addDal(double price) {
        System.out.println("Dal : "+price);
```

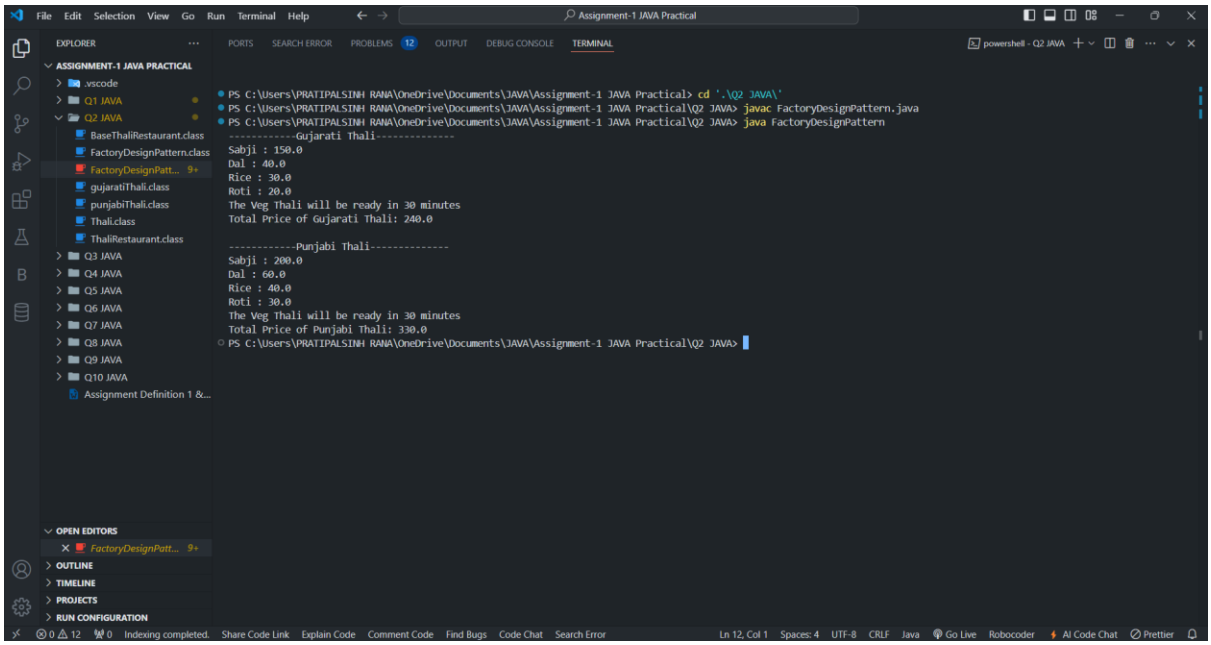
```
        setPrice(price);
    }

    public void addRice(double price) {
        System.out.println("Rice : "+price);
        setPrice(price);
    }

    public void addRoti(double price) {
        System.out.println("Roti : "+price);
        setPrice(price);
    }
}

class PunjabiThali extends Thali {

    public void addSabji(double price) {
        System.out.println("Sabji : "+price);
        setPrice(price);
    }

    public void addDal(double price) {
        System.out.println("Dal : "+price);
        setPrice(price);
    }
```

```
    public void addRice(double price) {
        System.out.println("Rice : "+price);
        setPrice(price);
    }

    public void addRoti(double price) {
        System.out.println("Roti : "+price);
        setPrice(price);
    }
}

abstract class BaseThaliRestaurant {
    public abstract Thali createThali(String type);
}

class ThaliRestaurant extends BaseThaliRestaurant {

    public Thali createThali(String type) {
        if (type.equals("Gujarati")) {
            return new GujaratiThali();
        } else if (type.equals("Punjabi")) {
            return new PunjabiThali();
        } else {
            return null;
        }
    }
```

```
}

class FactoryDesignPattern {

    public static void main(String[] args) {

        ThaliRestaurant restaurant = new ThaliRestaurant();

        System.out.println("------------Gujarati Thali--------------");
        Thali gujaratiThali = restaurant.createThali("Gujarati");
        gujaratiThali.addSabji(150.00);
        gujaratiThali.addDal(40.00);
        gujaratiThali.addRice(30.00);
        gujaratiThali.addRoti(20.00);
        gujaratiThali.makeThali();
        System.out.println("Total Price of Gujarati Thali: " +
gujaratiThali.getPrice());
        System.out.println("");

        System.out.println("------------Punjabi Thali--------------");
        Thali punjabiThali = restaurant.createThali("Punjabi");
        punjabiThali.addSabji(200.00);
        punjabiThali.addDal(60.00);
        punjabiThali.addRice(40.00);
        punjabiThali.addRoti(30.00);
        punjabiThali.makeThali();
```

```
        System.out.println("Total Price of Punjabi Thali: " +
punjabiThali.getPrice());
    }
}
```

Output:



Q.3:

```
import java.util.*;


interface PizzaOrderSystem{
    public void placeOrder(String pizzaType, int quantity);
    public String checkOrderStatus(int orderId);
    public boolean cancelOrder(int orderId);
    public double calculateOrderCost(int orderId);
    public void displayOrderDetails();
```

```
    public void listAvailablePizzas();
}

class PizzaOrderProcessor implements PizzaOrderSystem{
    int orderId=1;
    String[] listOfPizzas = {"Margherita Pizza","Cheese Pizza","Veggie Pizza","Non Veggie Pizza"};
    double[] listOfPizzasPrices = {90.00,150.00,100.45,400.85};
    String[][] orders = new String[5][3];

    public void placeOrder(String pizzaType, int quantity){
        System.out.println("Order ID "+this.orderId+" for Order of "+quantity+" "+pizzaType);
        orders[this.orderId-1][0] = Integer.toString(this.orderId);
        orders[this.orderId-1][1] = pizzaType;
        orders[this.orderId-1][2] = Integer.toString(quantity);
        this.orderId++;
    }

    public String checkOrderStatus(int orderId){

        for (int i = 0; i < orders.length; i++) {
            if(orderId == Integer.parseInt(orders[i][0])){
                return "New order placed successfully so it will be ready in 10 minutes";
            }
        }
```

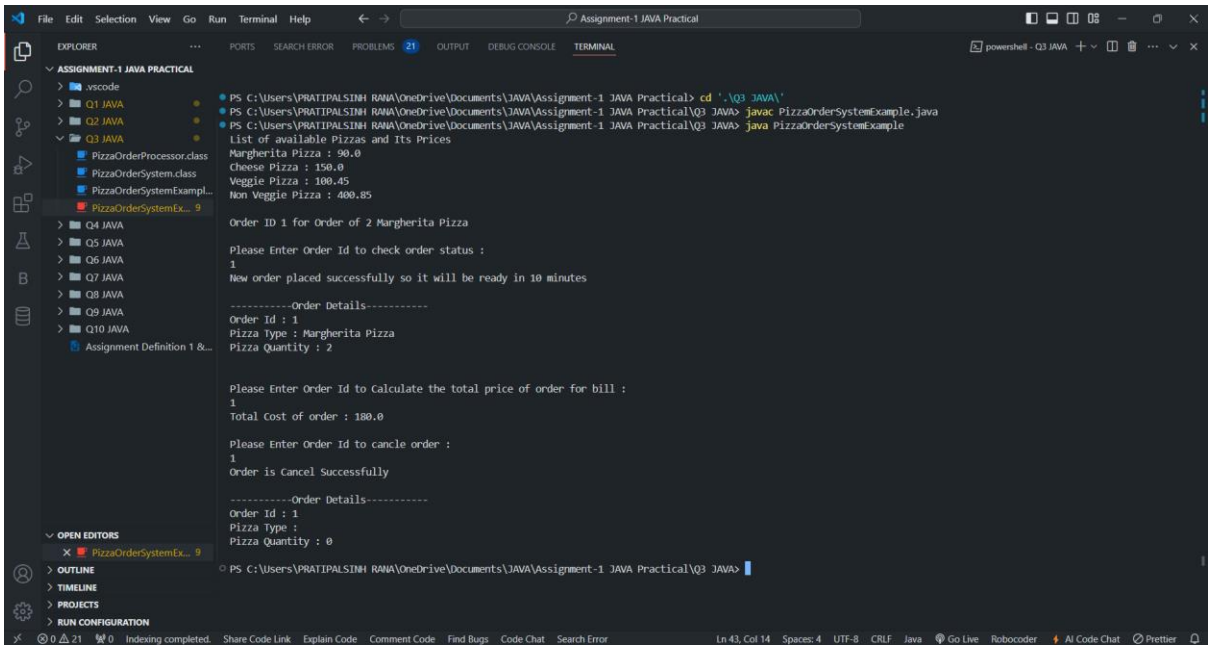
```
    return "Order is not placed";

}

public boolean cancelOrder(int orderId){
    for (int i = 0; i < this.orderId-1; i++) {
        if(orderId == Integer.parseInt(orders[i][0])){
            orders[orderId-1][1] = "";
            orders[orderId-1][2] = Integer.toString(0);
            return true;
        }
    }
    return false;
}

public double calculateOrderCost(int orderId){
        for (int i = 0; i < this.orderId-1; i++) {
            if(orderId == Integer.parseInt(orders[i][0])){
                return (getPrice(orderId)) * (Integer.parseInt(orders[i][2]));
            }
        }
        return 0.0;
}

public double getPrice(int orderId){
    for (int i = 0; i < listOfPizzas.length; i++) {
```

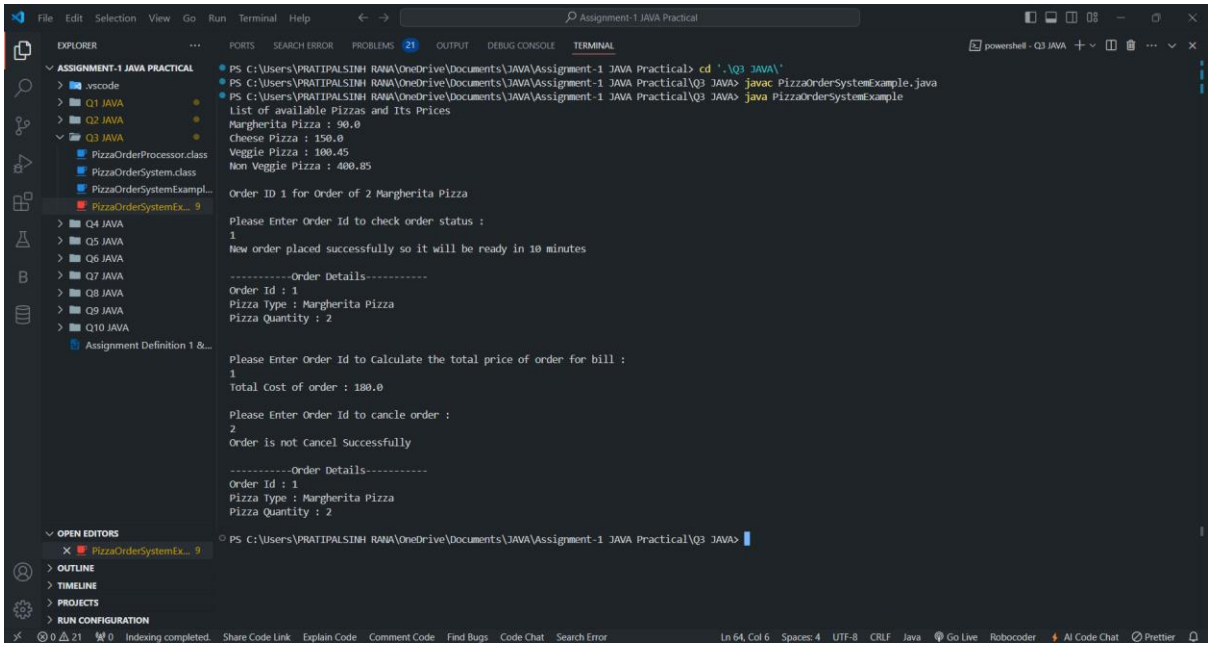
```
            if(orders[orderId-1][1] == listOfPizzas[i]){
                return listOfPizzasPrices[i];
            }
        }
        return 0.0;
    }

    public void displayOrderDetails(){
        System.out.println("\n----------Order Details----------");
        for (int i = 0; i < this.orderId-1; i++) {
            System.out.println("Order Id : "+orders[i][0]);
            System.out.println("Pizza Type : "+orders[i][1]);
            System.out.println("Pizza Quantity : "+orders[i][2]);
            System.out.println("");
        }
    }

    public void listAvailablePizzas(){
        System.out.println("List of available Pizzas and Its Prices");
        if((listOfPizzas.length) == (listOfPizzasPrices.length))
        {
            for (int i = 0; i < listOfPizzas.length; i++) {
                System.out.println(listOfPizzas[i]+" : "+listOfPizzasPrices[i]);
            }
        }
        System.out.println("");
```

```
    }

}
class PizzaOrderSystemExample{
    public static void main(String[] args){
        PizzaOrderSystem order1 = new PizzaOrderProcessor();
        Scanner s = new Scanner(System.in);
        order1.listAvailablePizzas();

        order1.placeOrder("Margherita Pizza", 2);

        System.out.println("\nPlease Enter Order Id to check order status : ");
        int orderid = s.nextInt();
        System.out.println(order1.checkOrderStatus(orderid));

        order1.displayOrderDetails();

        System.out.println("\nPlease Enter Order Id to Calculate the total price of
order for bill : ");
        orderid = s.nextInt();
        System.out.println("Total Cost of order :
"+order1.calculateOrderCost(orderid));

        System.out.println("\nPlease Enter Order Id to cancle order : ");
        orderid = s.nextInt();
        if(order1.cancelOrder(orderid)){
            System.out.println("Order is Cancel Successfully");
```

```
        }
        else{
            System.out.println("Order is not Cancel Successfully");
        }

        order1.displayOrderDetails();
    }
}
```

Output:





Q.4:

```
class Person{
    String name;
    int age;

    Person(String name, int age){
        this.name = name;
        setAge(age);
    }

    String getName(){
        return this.name;
    }

    void setName(String name){
```

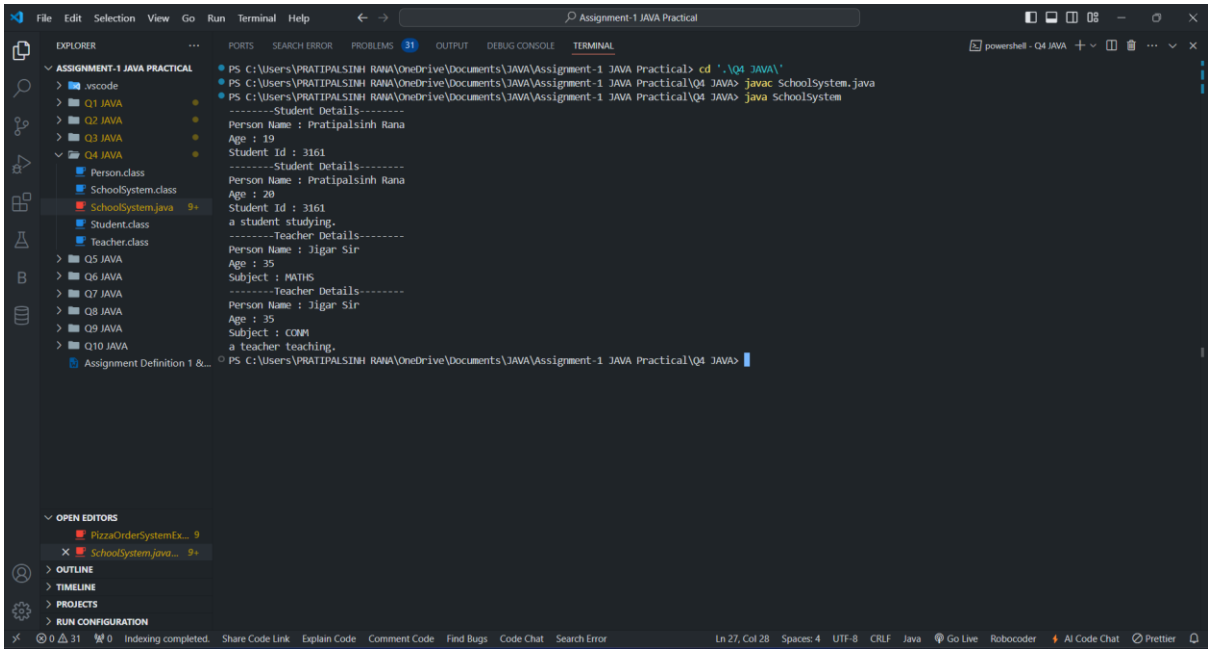
```
        this.name = name;
    }

    int getAge(){
        return this.age;
    }

    void setAge(int age){
        if(age < 0){
            System.out.println("Please Enter age in positive value");
        }
        else{
            this.age = age;
        }
    }

    void introduce(){
        System.out.println("Person Name : "+name+" \nAge : "+age);
    }
}

class Student extends Person{
    int studentId;

    Student(String name, int age,int studentId) {
        super(name, age);
```

```
        this.studentId = studentId;
    }

    int getStudentId(){
        return studentId;
    }

    void setStudentId(int studentId){
        this.studentId = studentId;
    }

    void introduce(){
        System.out.println("--------Student Details--------");
        super.introduce();
        System.out.println("Student Id : "+studentId);
    }

    void study(){
        System.out.println("a student studying.");
    }

}

class Teacher extends Person{
    String subject;
```

```
    Teacher(String name, int age, String subject) {
        super(name, age);
        this.subject = subject;
    }

    String getSubject(){
        return this.subject;
    }

    void setSubject(String subject){
        this.subject = subject;
    }

    void introduce(){
        System.out.println("--------Teacher Details--------");
        super.introduce();
        System.out.println("Subject : "+subject);
    }

    void teach(){
        System.out.println("a teacher teaching.");
    }
}
class SchoolSystem{
    public static void main(String[] args){
        Student s1 = new Student("Pratipalsinh Rana",19,3161);
```

```
        Teacher t1 = new Teacher("Jigar Sir",35,"MATHS");
        s1.introduce();
        s1.setAge(20);
        s1.introduce();
        s1.study();

        t1.introduce();
        t1.setSubject("CONM");
        t1.introduce();
        t1.teach();
    }
}
```

Output:



Q.5:

```
class User{
   String username;
   String email;

   User(String username, String email){
      this.username = username;
      this.email = email;
   }

   String getUsername(){
      return this.username;
   }

   String getEmail(){
      return this.email;
   }

}

class Professor extends User{
   String department;

   Professor(String username, String email,String department){
      super(username,email);
      this.department = department;
```

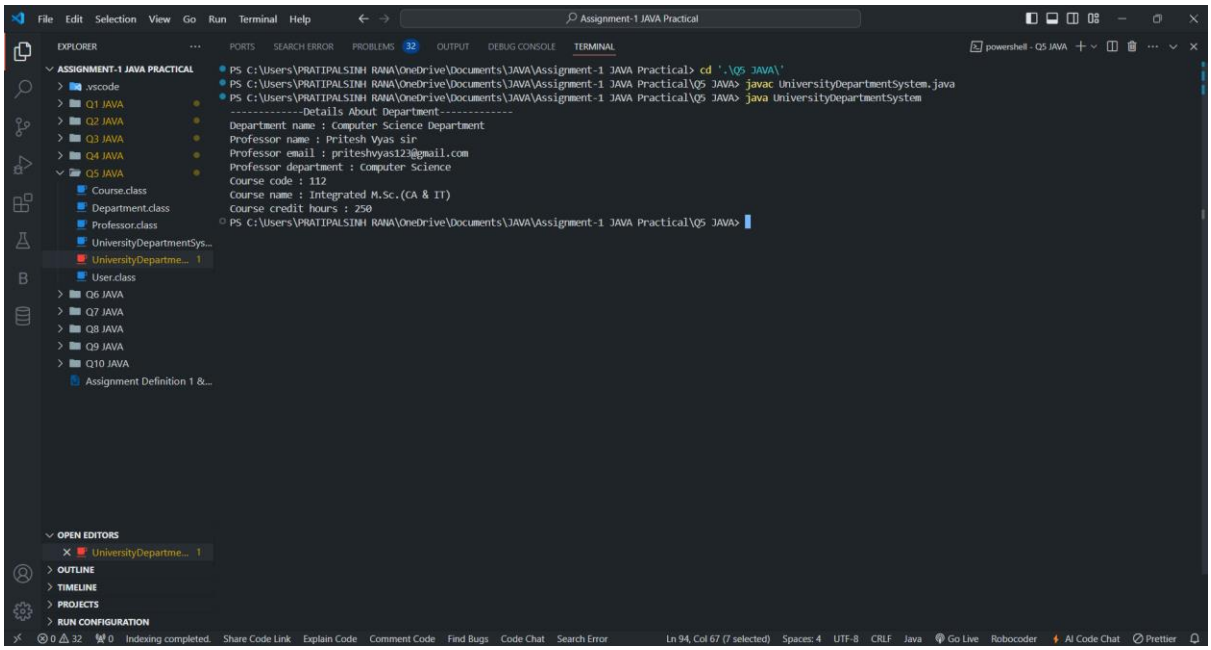
```
    }

    String getDepartment(){
        return this.department;
    }

}

class Course{
    int code;
    String courseName;
    int creditHours;

    Course(int code, String courseName, int creditHours){
        this.code = code;
        this.courseName = courseName;
        this.creditHours = creditHours;
    }

    int getCode(){
        return this.code;
    }

    String getCoursename(){
        return this.courseName;
    }
```

```
    int getCreditHours(){
        return this.creditHours;
    }
}

class Department{
    String departmentName;
    Professor professor1;
    Course course1;

    Department(String departmentName){
        this.departmentName = departmentName;
    }

    String getDepartmentname(){
        return this.departmentName;
    }

    Professor getProfessor1(){
        return professor1;
    }

    void setProfessor1(Professor professor1){
        this.professor1 = professor1;
    }
```

```
    Course getCourse1(){
        return course1;
    }

    void setCourse1(Course course1){
        this.course1 = course1;
    }
}

class UniversityDepartmentSystem{
    public static void main(String[] args) {
        Professor p1 = new Professor("Pritesh Vyas
sir","priteshvyas123@gmail.com","Computer Science");

        Course c1 = new Course(112,"Integrated M.Sc.(CA & IT)",250);

        Department csDepartment = new Department("Computer Science
Department");

        csDepartment.setProfessor1(p1);
        csDepartment.setCourse1(c1);

        System.out.println("-------------Details About Department-------------");
        System.out.println("Department name :
"+csDepartment.getDepartmentname());
```

```
        System.out.println("Professor name :
"+csDepartment.getProfessor1().getUsername());

        System.out.println("Professor email :
"+csDepartment.getProfessor1().getEmail());

        System.out.println("Professor department :
"+csDepartment.getProfessor1().getDepartment());

        System.out.println("Course code :
"+csDepartment.getCourse1().getCode());

        System.out.println("Course name :
"+csDepartment.getCourse1().getCoursename());

        System.out.println("Course credit hours :
"+csDepartment.getCourse1().getCreditHours());


    }
}
```

Output:



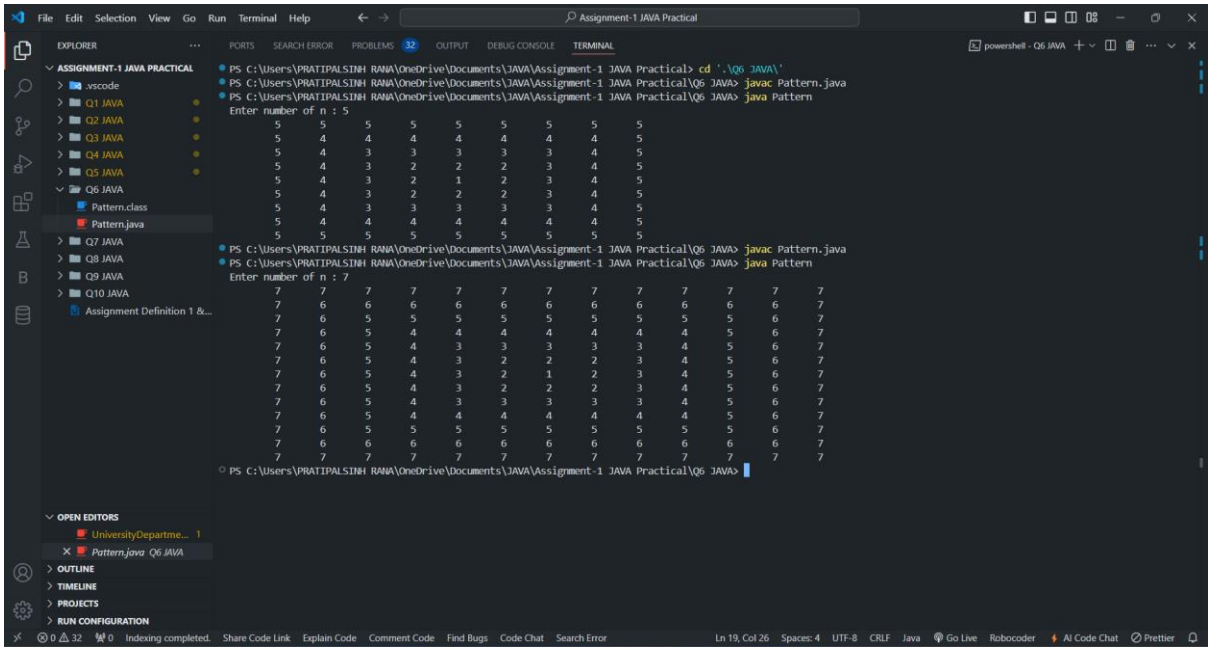
Q.6:

```
import java.util.Scanner;


class Pattern{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);


        System.out.print("Enter number of n : ");
        int n = s.nextInt();


        int size = (n*2)-1;
        int arr[][] = new int[size][size];


        int top = 0;
        int bottom = size-1;
        int left = 0;
        int right = size-1;


        while (n != 0) {
            // top-left to top-right
            for (int i = left; i < right; i++) {
                arr[top][i] = n;
            }


            //right-top to right-bottom
            for (int j = top; j < bottom; j++) {
```

```
            arr[j][right] = n;
        }

        //bottom-right to bottom-left
        for (int k = right; k >= left; k--) {
            arr[bottom][k] = n;
        }

        //left-bottom to left-top
        for (int l = bottom; l >= top; l--) {
            arr[l][left] = n;
        }
        top++;
        bottom--;
        left++;
        right--;
        n--;
    }

    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            System.out.print("\t"+arr[i][j]);
        }
        System.out.println("");
    }
}
```

}

Output:

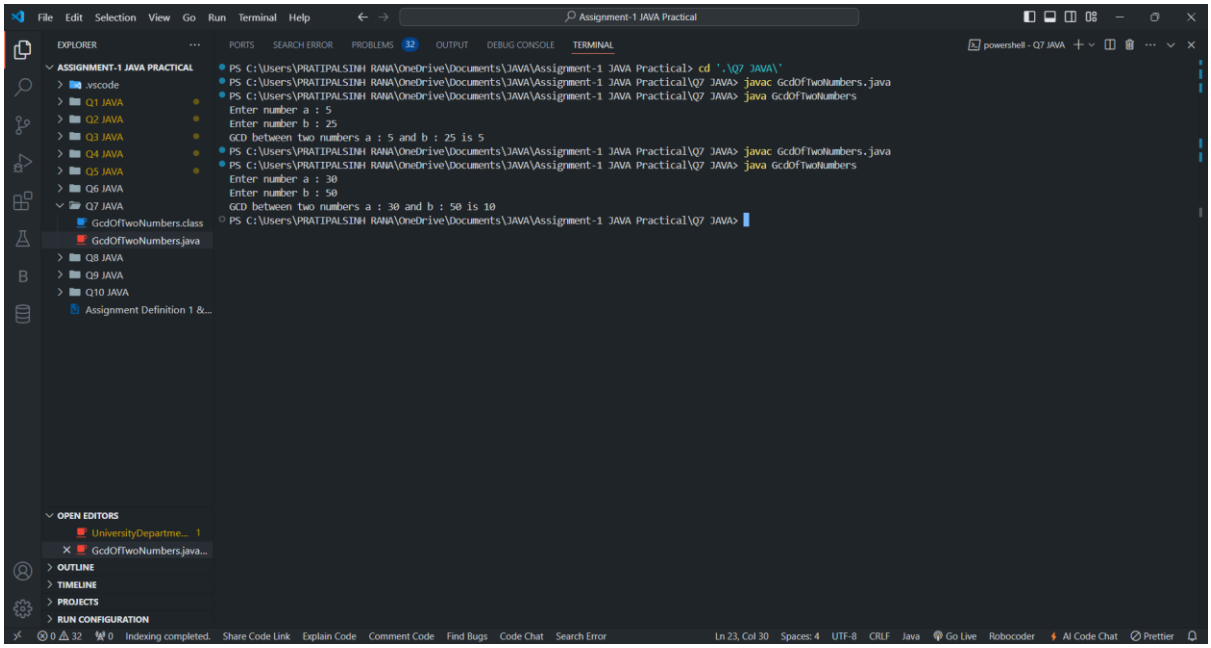


Q.7:

```
import java.util.Scanner;


class GcdOfTwoNumbers{
    public static void main(String[] args) {
        int a;
        int b;


        Scanner s = new Scanner(System.in);
        System.out.print("Enter number a : ");
        a = s.nextInt();
```

```
        System.out.print("Enter number b : ");
        b = s.nextInt();

        System.out.println("GCD between two numbers a : "+a+" and b : "+b+" is
"+findGcd(a,b));
    }

    private static int findGcd(int x, int y){
        int SmallerValue;
        if (x < y) {
            SmallerValue = x;
        }
        else{
            SmallerValue = y;
        }

        int i = SmallerValue;

        while (i > 1) {
            if(x % i == 0 && y % i == 0){
                return i;
            }
            i--;
        }
        return 1;
    }
}
```

Output:



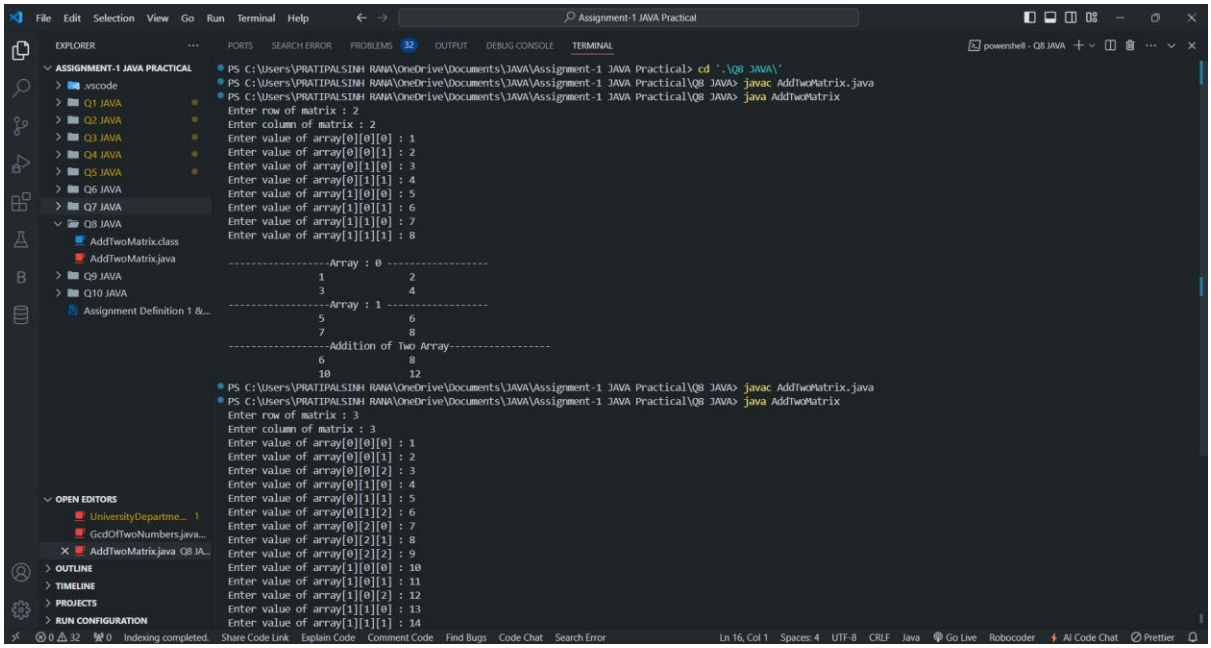
Q.8:

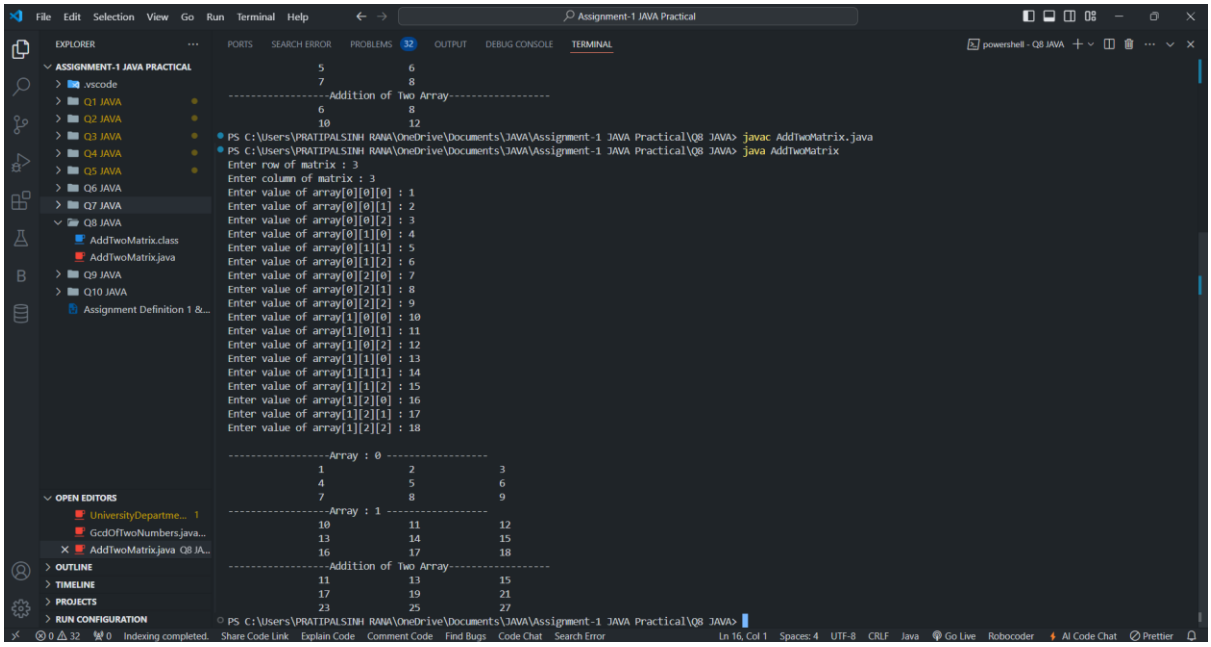
```
import java.util.Scanner;


class AddTwoMatrix{
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);


        System.out.print("Enter row of matrix : ");
        int row = s.nextInt();
```

```
System.out.print("Enter column of matrix : ");
int column = s.nextInt();

int[][][] array = new int[2][row][column];

for (int k = 0; k < 2; k++) {
   for (int i = 0; i < row; i++) {
      for (int j = 0; j < column; j++) {
         System.out.print("Enter value of array["+k+"]["+i+"]["+j+"] : ");
         array[k][i][j]=s.nextInt();
      }
   }
}

int[][] arraysSum = new int[row][column];

for (int k = 0; k < 2-1; k++) {
   for (int i = 0; i < row; i++) {
      for (int j = 0; j < column; j++) {
         arraysSum[i][j] = array[k][i][j] + array[k+1][i][j];
      }
   }
}

System.out.println("");
for (int k = 0; k < 2; k++) {
```

```
            System.out.println("------------------Array : "+k+" ------------------");
            for (int i = 0; i < row; i++) {
                for (int j = 0; j < column; j++) {
                    System.out.print("\t\t"+array[k][i][j]);
                }
                System.out.println("");
            }
        }

        System.out.print("------------------Addition of Two Array------------------");
        for (int i = 0; i < row; i++) {
            System.out.println("");
            for (int j = 0; j < column; j++) {
                    System.out.print("\t\t"+arraysSum[i][j]);
            }
        }
    }
}
```

Output:





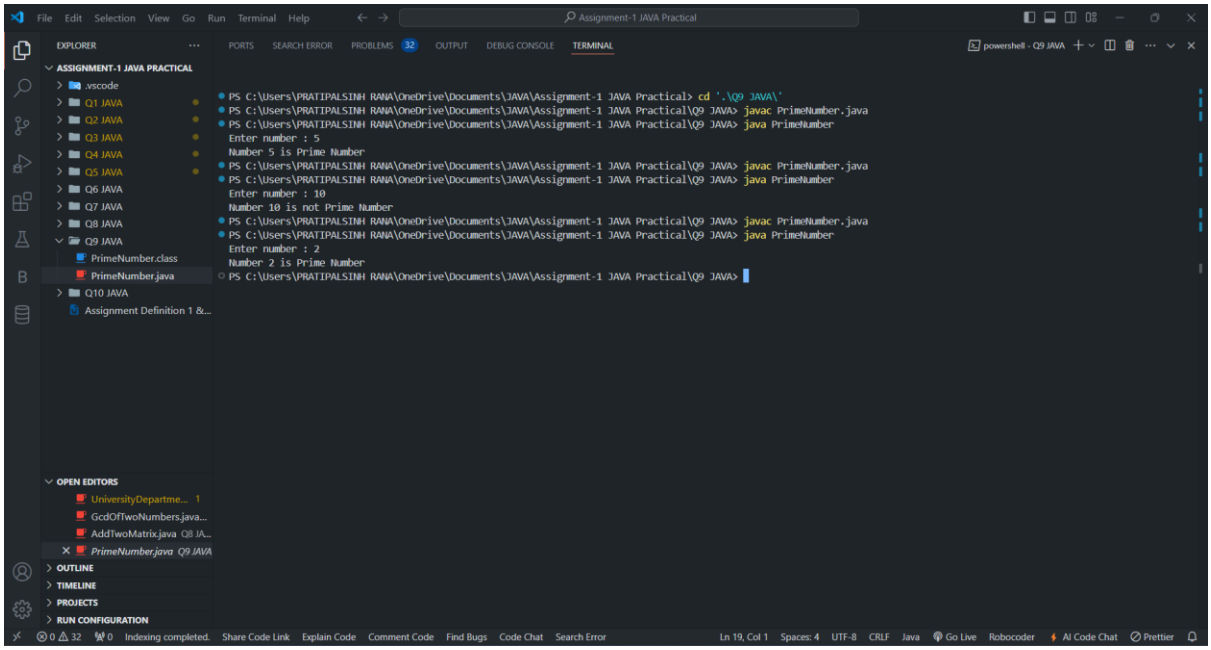
Q.9:

```
import java.util.Scanner;


class PrimeNumber{
    public static void main(String[] args){
```

```
    Scanner s = new Scanner(System.in);

    System.out.print("Enter number : ");
    int n = s.nextInt();

    if(chekcisPrimeNumber(n)){
       System.out.println("Number "+n+" is Prime Number");
    }
    else{
       System.out.println("Number "+n+" is not Prime Number");
    }
}

public static boolean chekcisPrimeNumber(int n){
    int i = 2;

    if (n <= 2) {
       return n == 2;
    }

    if(n % i == 0){
       return false;
    }

    if(i < Math.sqrt(n)){
       return true;
```

```
        }


        i++;
        return chekcisPrimeNumber(n);
    }
}
```

Output:



Q.10:

```
class StringBufferExample{
    public static void main(String[] args) {
        StringBuffer stringBuffer = new StringBuffer("Hello, World!");
        System.out.println("String : "+stringBuffer);
```

```
        stringBuffer.append(" Welcome to Java!");
        System.out.println("String after using append method : "+stringBuffer);

        stringBuffer.insert(12, "from ");
        System.out.println("String after using insert method : "+stringBuffer);

        stringBuffer.replace(7, 12, "Universe");
        System.out.println("String after using replace method : "+stringBuffer);

        stringBuffer.setCharAt(0, 'h');
        System.out.println("String after using setCharAt method : "+stringBuffer);

        stringBuffer.delete(2, 5);
        System.out.println("String after using delete method : "+stringBuffer);

        stringBuffer.deleteCharAt(10);
        System.out.println("String after using deleteCharAt method :
"+stringBuffer);

        stringBuffer.reverse();
        System.out.println("String after using reverse method : "+stringBuffer);

        String result = stringBuffer.toString();
        System.out.println("String after using toString method : "+stringBuffer);

        int length = stringBuffer.length();
        int capacity = stringBuffer.capacity();
```

```
        System.out.println("Modified String: " + result);
        System.out.println("Length of StringBuffer: " + length);
        System.out.println("Capacity of StringBuffer: " + capacity);
    }
}
```

Output: