ASSIGNMENT – 1

Subject: CSW2 (CSE 2141)

Name: Arpit Kumar Mohanty

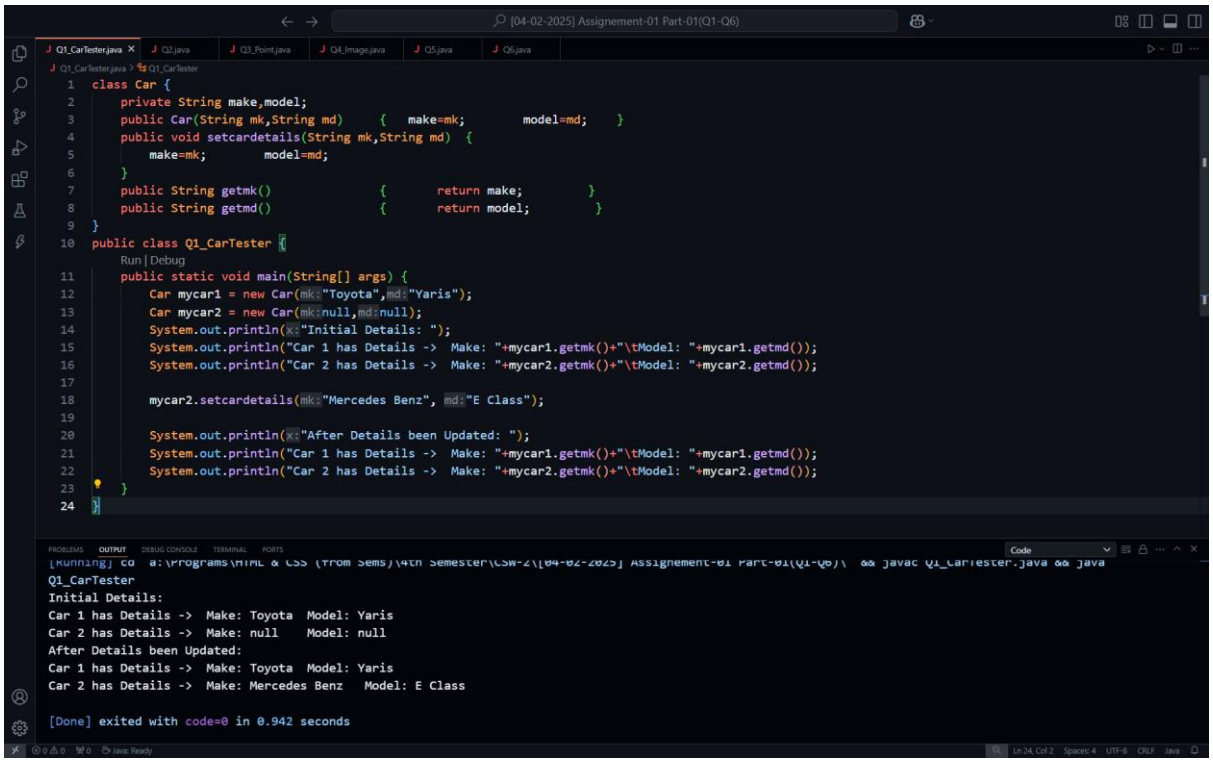
Registration Number: 2341013237

Section: 23412G1

Branch: CSE

Q1. Write a Java program with a Car class having private fields (make, model), a parameterized constructor, getter, and setter methods. In the CarTester class, instantiate myCar1 with values and myCar2 with null. Print their initial details, update myCar2 using setters, and print the updated details.

Solution along with Output:



```
class Car {
    private String make,model;
    public Car(String mk,String md)     {   make=mk;        model=md;    }
    public void setcardetails(String mk,String md)  {
        make=mk;        model=md;
    }
    public String getmk()               {       return make;        }
    public String getmd()               {       return model;        }
}
public class Q1_CarTester {
    Run | Debug
    public static void main(String[] args) {
        Car mycar1 = new Car(mk:"Toyota",md:"Yaris");
        Car mycar2 = new Car(mk:null,md:null);
        System.out.println(x:"Initial Details: ");
        System.out.println("Car 1 has Details ->  Make: "+mycar1.getmk()+"\tModel: "+mycar1.getmd());
        System.out.println("Car 2 has Details ->  Make: "+mycar2.getmk()+"\tModel: "+mycar2.getmd());

        mycar2.setcardetails(mk:"Mercedes Benz", md:"E Class");

        System.out.println(x:"After Details been Updated: ");
        System.out.println("Car 1 has Details ->  Make: "+mycar1.getmk()+"\tModel: "+mycar1.getmd());
        System.out.println("Car 2 has Details ->  Make: "+mycar2.getmk()+"\tModel: "+mycar2.getmd());
    }
}
```

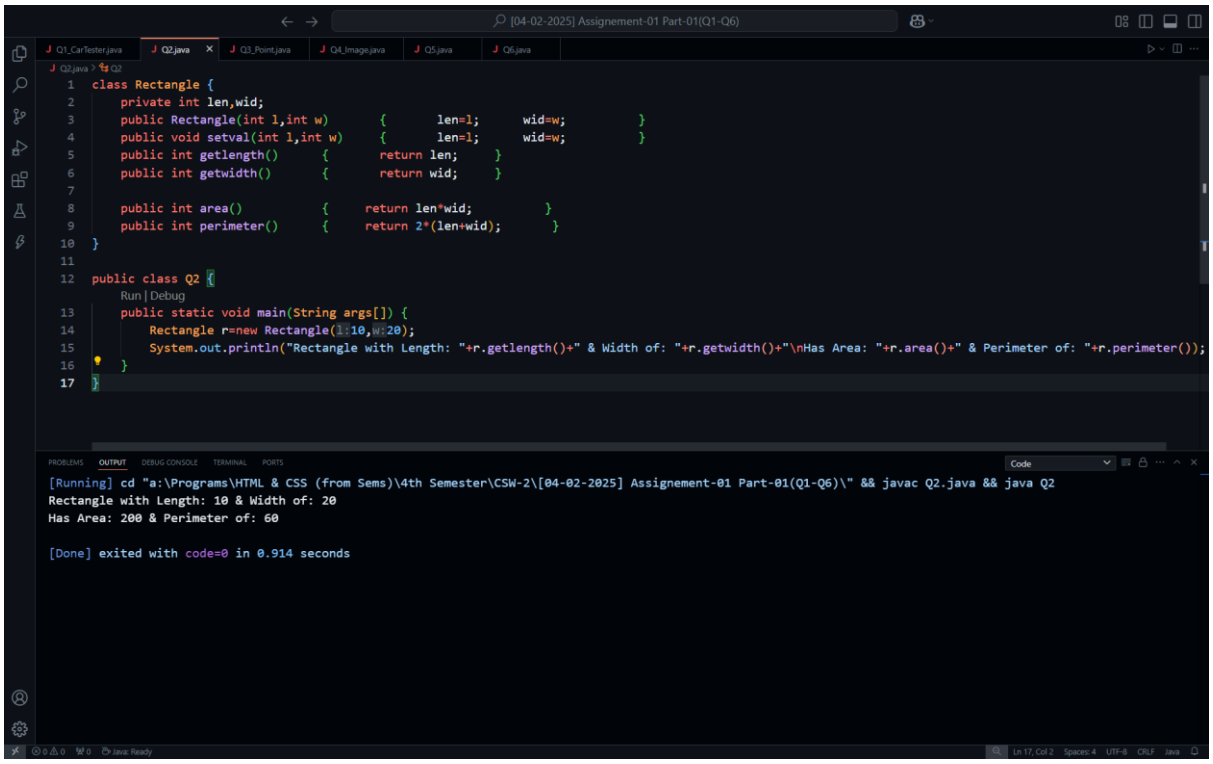
```
Q1_CarTester
Initial Details:
Car 1 has Details ->  Make: Toyota  Model: Yaris
Car 2 has Details ->  Make: null    Model: null
After Details been Updated:
Car 1 has Details ->  Make: Toyota  Model: Yaris
Car 2 has Details ->  Make: Mercedes Benz   Model: E Class

[Done] exited with code=0 in 0.942 seconds
```

Q2. Design a Java class called Rectangle with private attributes length and width. Include a constructor to initialize these attributes, as well as setter and getter methods for each attribute. Additionally, implement

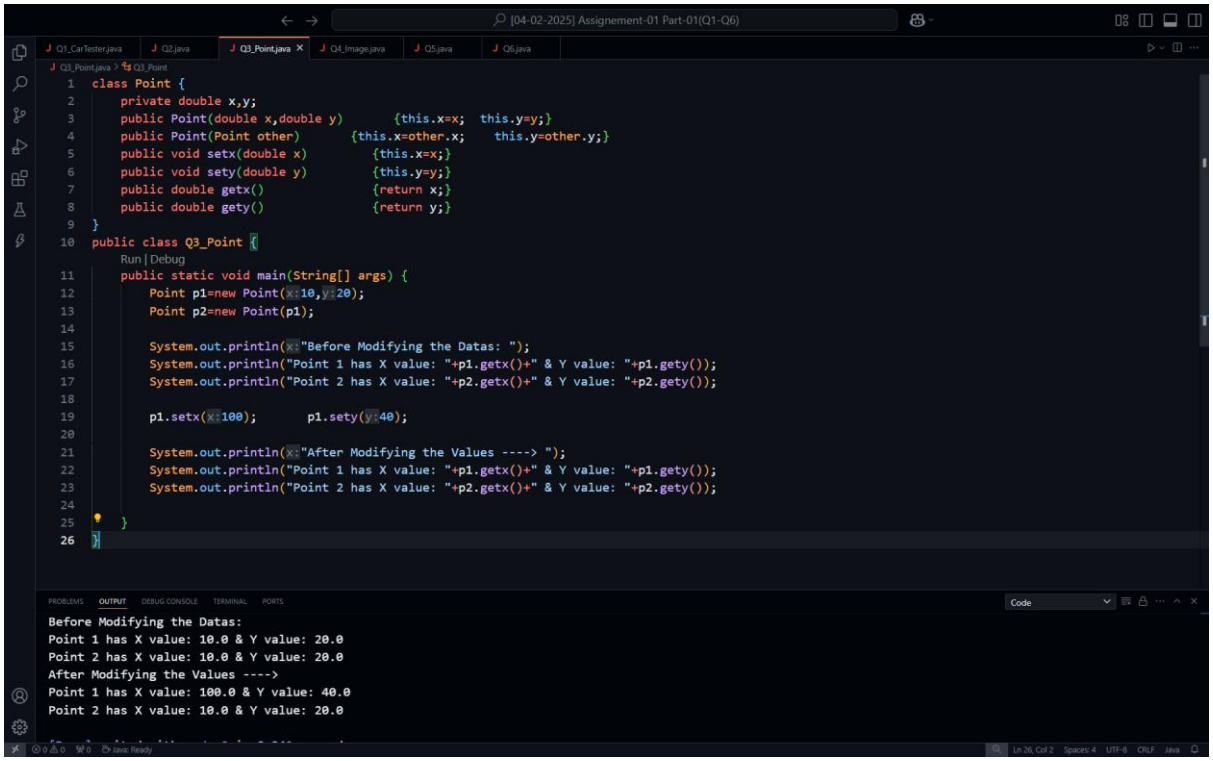
methods to calculate the area and perimeter of the rectangle. Write a main method to create an object of the Rectangle class, set values for its attributes, and display the area and perimeter.

Solution along with Output:



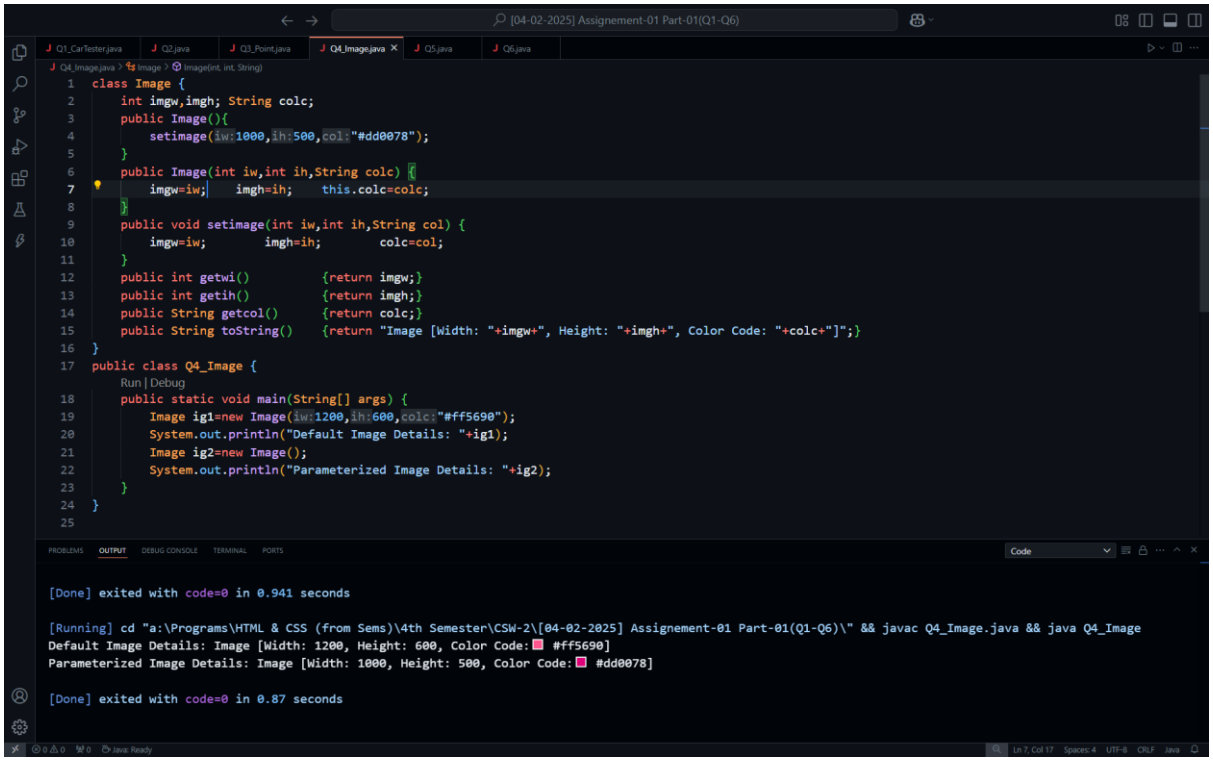
Q3. Write a Java program that defines a Point class with attributes X and Y, and includes a parameterized constructor to initialize these attributes. Implement a copy constructor to create a new point object with the same attribute values. Ensure that modifications made to one object do not affect the other. Utilize getter and setter methods to retrieve and update the attribute values.

Solution along with Output:



Q4. Write a program to create an Image class with attributes imageWidth, imageHeight, and colorCode. Add the required constructor, set methods, get methods, and toString method. Create the object of the image class using the default and parameterized constructor and print the details of the object.

Solution along with Output:



Q5. Create an ImageLibrary, which contains a set of image objects (from Q4) and operations such as searching an image, inserting an image, and getting an image. Create an ImageController class to manage the program execution and call the methods to create and manipulate images.

Solution:

```java
import java.util.*;
class ImageController {
    HashSet<Image> ImgLib = new HashSet<>();
    public void insert(Image I) {
        ImgLib.add(I);
    }
    public void Search(Image I) {
        if (ImgLib.contains(I)) {
            System.out.println(x:"The Image has been found ");
            System.out.println(I);
        }
        else {
            System.out.println(x:"Image is not found.");
        }
    }
    public void getImage(Image I) {
        System.out.println(I);
    }
}
public class Q5 {
    Run | Debug
    public static void main(String[] args) {
        Image I2 = new Image(iw:300, ih:400, colc:"#fff666");
        Image I3 = new Image(iw:200, ih:300, colc:"#eee450");
        Image I4 = new Image(iw:150, ih:600, colc:"#3480fe");
        ImageController IC = new ImageController();
        IC.insert(I2);
        IC.insert(I4);
        IC.getImage(I2);
        IC.Search(I3);
        IC.Search(I4);
    }

}
```

Output:

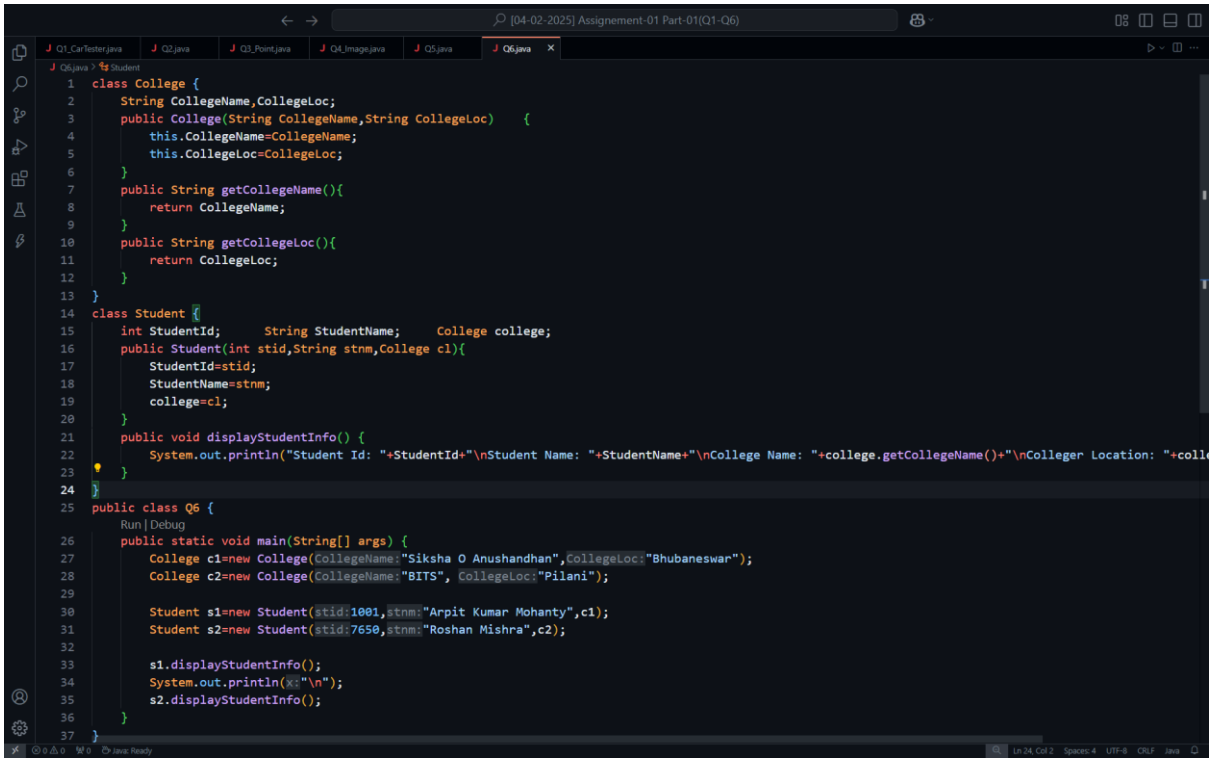
```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semes
Image [Width: 300, Height: 400, Color Code: #fff666]
Image is not found.
The Image has been found
Image [Width: 150, Height: 600, Color Code: #3480fe]

[Done] exited with code=0 in 0.948 seconds
```

Q6. Develop a Java-based College Management System to model the relationship between colleges and students. Create a College class with attributes collegeName and collegeLoc, and a Student class with studentId, studentName, and a reference to a College object. Implement a constructor in Student to initialize these attributes and a displayStudentInfo() method to print student and college details. In the MainApp class, instantiate at least two College and Student objects

enroll each student in one of the colleges, and display all details using appropriate methods.

Solution:



```java
class College {
    String CollegeName,CollegeLoc;
    public College(String CollegeName,String CollegeLoc)    {
        this.CollegeName=CollegeName;
        this.CollegeLoc=CollegeLoc;
    }
    public String getCollegeName(){
        return CollegeName;
    }
    public String getCollegeLoc(){
        return CollegeLoc;
    }
}
class Student {
    int StudentId;      String StudentName;     College college;
    public Student(int stid,String stnm,College cl){
        StudentId=stid;
        StudentName=stnm;
        college=cl;
    }
    public void displayStudentInfo() {
        System.out.println("Student Id: "+StudentId+"\nStudent Name: "+StudentName+"\nCollege Name: "+college.getCollegeName()+"\nColleger Location: "+coll
    }
}
public class Q6 {
    Run | Debug
    public static void main(String[] args) {
        College c1=new College(CollegeName:"Siksha O Anushandhan",CollegeLoc:"Bhubaneswar");
        College c2=new College(CollegeName:"BITS", CollegeLoc:"Pilani");

        Student s1=new Student(stid:1001,stnm:"Arpit Kumar Mohanty",c1);
        Student s2=new Student(stid:7650,stnm:"Roshan Mishra",c2);

        s1.displayStudentInfo();
        System.out.println(x:"\n");
        s2.displayStudentInfo();
    }
}
```

Output:

```
[Running] cd "a:\Programs\HTML & CSS (from Sems)\4th Semester\CSW-2\[04-0
Student Id: 1001
Student Name: Arpit Kumar Mohanty
College Name: Siksha O Anushandhan
Colleger Location: Bhubaneswar


Student Id: 7650
Student Name: Roshan Mishra
College Name: BITS
Colleger Location: Pilani

[Done] exited with code=0 in 0.858 seconds
```