

## LAB # 14

### Mutability and Immutability

#### **OBJECTIVE:**

Understanding and implementing the concept of mutability and immutability.

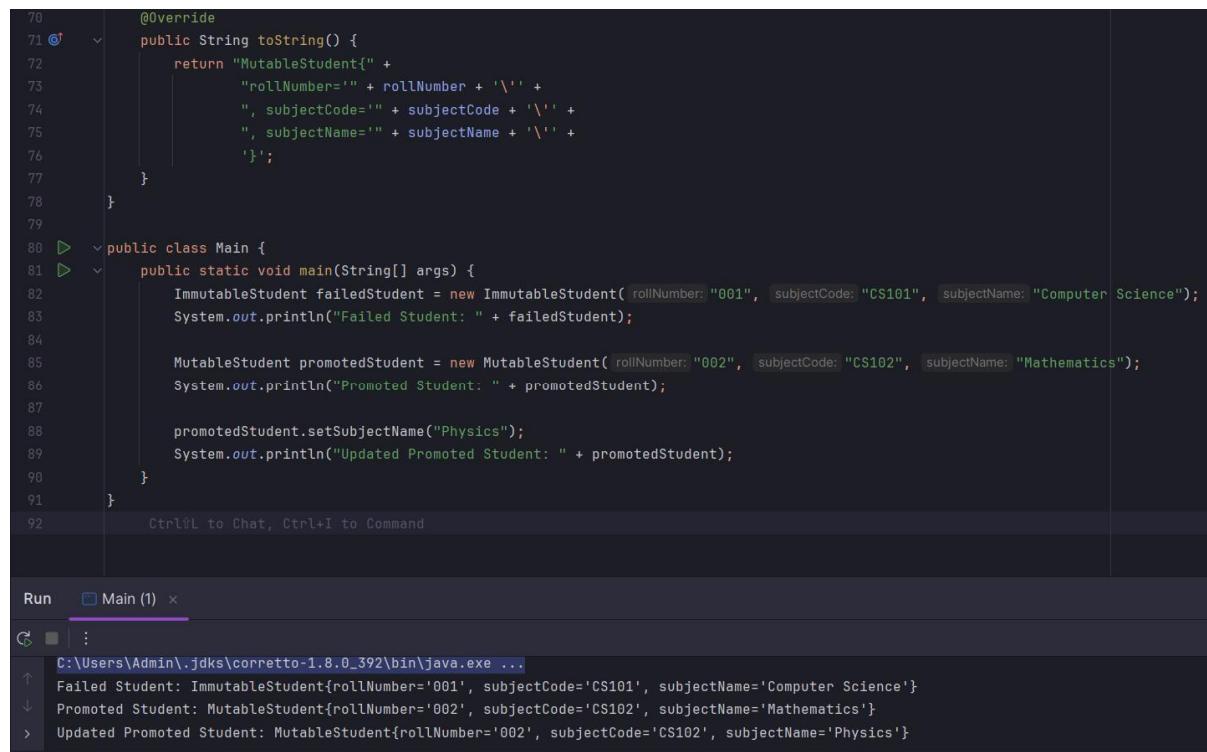
#### **LAB TASK:**

1. Apply concept of mutability and immutability for the task promoted and failed students.

The roll number, subject code, and subject name would have to be entered at time of object creation and with getter method these variables should be printed. (Hint: Those students who are failed in previous semester will be registered in immutable class, and promoted students are registered in mutable class)

```
 2 usages
1  final class ImmutableStudent {
2      3 usages
3          private final String rollNumber;
4          3 usages
5              private final String subjectCode;
6              3 usages
7                  private final String subjectName;
8
9                  1 usage
10                 public ImmutableStudent(String rollNumber, String subjectCode, String subjectName) {
11                     this.rollNumber = rollNumber;
12                     this.subjectCode = subjectCode;
13                     this.subjectName = subjectName;
14                 }
15
16                 no usages
17                 public String getRollNumber() {
18                     return rollNumber;
19                 }
20
21                 no usages
22                 public String getSubjectCode() {
23                     return subjectCode;
24                 }
25
26                 no usages
27                 public String getSubjectName() {
28                     return subjectName;
29                 }
30
31             @Override
32             public String toString() {
```

```
23
24
25 @Override
26     public String toString() {
27         return "ImmutableStudent{" +
28             "rollNumber='" + rollNumber + '\'' +
29             ", subjectCode='" + subjectCode + '\'' +
30             ", subjectName='" + subjectName + '\'' +
31             '}';
32     }
33
34
35     2 usages
36     class MutableStudent {
37         4 usages
38         private String rollNumber;
39         4 usages
40         private String subjectCode;
41         4 usages
42         private String subjectName;
43
44         1 usage
45         public MutableStudent(String rollNumber, String subjectCode, String subjectName) {
46             this.rollNumber = rollNumber;
47             this.subjectCode = subjectCode;
48             this.subjectName = subjectName;
49         }
50
51         no usages
52         public String getRollNumber() {
53             return rollNumber;
54         }
55
56         no usages
57         public void setRollNumber(String rollNumber) {
58             this.rollNumber = rollNumber;
59         }
60
61         no usages
62         public String getSubjectCode() {
63             return subjectCode;
64         }
65
66         no usages
67         public void setSubjectCode(String subjectCode) {
68             this.subjectCode = subjectCode;
69         }
70
71         no usages
72         public String getSubjectName() {
73             return subjectName;
74         }
75
76         1 usage
77         public void setSubjectName(String subjectName) {
78             this.subjectName = subjectName;
79         }
80
81     }
```



```

70     @Override
71     public String toString() {
72         return "MutableStudent{" +
73             "rollNumber='" + rollNumber + '\'' +
74             ", subjectCode='" + subjectCode + '\'' +
75             ", subjectName='" + subjectName + '\'' +
76             '}';
77     }
78 }
79
80 public class Main {
81     public static void main(String[] args) {
82         ImmutableStudent failedStudent = new ImmutableStudent(rollNumber: "001", subjectCode: "CS101", subjectName: "Computer Science");
83         System.out.println("Failed Student: " + failedStudent);
84
85         MutableStudent promotedStudent = new MutableStudent(rollNumber: "002", subjectCode: "CS102", subjectName: "Mathematics");
86         System.out.println("Promoted Student: " + promotedStudent);
87
88         promotedStudent.setSubjectName("Physics");
89         System.out.println("Updated Promoted Student: " + promotedStudent);
90     }
91 }
92 
```

Ctrl+L to Chat, Ctrl+I to Command

Run Main (1)

```

C:\Users\Admin\.jdks\corretto-1.8.0_392\bin\java.exe ...
↑ Failed Student: ImmutableStudent{rollNumber='001', subjectName='Computer Science'}
↓ Promoted Student: MutableStudent{rollNumber='002', subjectCode='CS102', subjectName='Mathematics'}
> Updated Promoted Student: MutableStudent{rollNumber='002', subjectCode='CS102', subjectName='Physics'}
```

2. Write a program that will calculate the below 4 formulas. Decide what to make mutable and what to make immutable and perform task operations. Formulas are:

Circumference of circle:  $C = 2 \pi r$

Area of circle:  $A = \pi r^2$

Volume of sphere:  $V = \frac{4}{3} \pi r^3$

Surface area of sphere:  $A = 4 \pi r^2$

(Hint: Value of pie would be constant and value of radius should be variant)

```

public class CircleSphereCalculations {
    4 usages
    private static final double PI = 3.14159;
    9 usages
    double radius;
    1 usage
    public CircleSphereCalculations(double radius) {
        this.radius = radius;
    }
    1 usage
    public void setRadius(double radius) {
        this.radius = radius;
    }
    no usages
    public double getRadius() {
        return radius;
    }
    2 usages
    public double calculateCircumference() {
        return 2 * PI * radius;
    }
    2 usages
    public double calculateAreaOfCircle() {
        return PI * radius * radius;
    }
    2 usages
    public double calculateVolumeOfSphere() {
        return (4.0 / 3.0) * PI * Math.pow(radius, 3);
    }
    2 usages
    public double calculateSurfaceAreaOfSphere() {
        return 4 * PI * radius * radius;
    }
}
```

```
        return PI * radius * radius;
    }
2 usages
public double calculateVolumeOfSphere() {
    return (4.0 / 3.0) * PI * Math.pow(radius, 3);
}
2 usages
public double calculateSurfaceAreaOfSphere() {
    return 4 * PI * radius * radius;
}

public static void main(String[] args) {
    CircleSphereCalculations calculator = new CircleSphereCalculations( radius: 5.0);

    System.out.println("Circumference of Circle: " + calculator.calculateCircumference());
    System.out.println("Area of Circle: " + calculator.calculateAreaOfCircle());
    System.out.println("Volume of Sphere: " + calculator.calculateVolumeOfSphere());
    System.out.println("Surface Area of Sphere: " + calculator.calculateSurfaceAreaOfSphere());

    calculator.setRadius(10.0);
    System.out.println("\nAfter updating radius to 10.0:");
    System.out.println("Circumference of Circle: " + calculator.calculateCircumference());
    System.out.println("Area of Circle: " + calculator.calculateAreaOfCircle());
    System.out.println("Volume of Sphere: " + calculator.calculateVolumeOfSphere());
    System.out.println("Surface Area of Sphere: " + calculator.calculateSurfaceAreaOfSphere());
}
}
```

The screenshot shows the run output of a Java application named 'CircleSphereCalculations'. The terminal window displays the following text:

```
C:\Users\Admin\.jdks\corretto-1.8.0_392\bin\java.exe ...
Circumference of Circle: 31.4159
Area of Circle: 78.53975
Volume of Sphere: 523.5983333333332
Surface Area of Sphere: 314.159

After updating radius to 10.0:
Circumference of Circle: 62.8318
Area of Circle: 314.159
Volume of Sphere: 4188.786666666666
Surface Area of Sphere: 1256.636

Process finished with exit code 0
```