PRACTICAL 5 (WEEK 6)

Faysal Hossain [0357986]

Part 1 — Implementing a Class

1. **Createing Student File :**

```
class Student {
    String name;

    Student(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

2. **Beginning comments**

```
/*
@(#)Student.java 1.0 2023/10/13
(c) Taylor's University, Malaysia
*/
```

```java
class Student {
    String name;

    Student(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

3. **Package and import statements**

    - The class doesn't necessitate any package or import declarations.

4. **javadoc comment for class**

```java
/**
 * Represents a student.
 * @version 1.0 13 Oct 2023
 * @author Faysal Hossain
 */
class Student {
    String name;

    Student(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

5. **Class definition**

code :

```java
public class Student {
    String name;

    Student(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

Compiling the File :

```
javac Student.java
```

Checking into CVS :

```
cvs add Student.java
```

6. **Instance variables**

```java
public class Student {
    private String name;

    Student(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

7. **Constructor**

```java
public class Student {
    private String name;

    public Student() {
        name = "dummy value";
    }

    Student(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

8. **Method**

```java
public class Student {
    private String name;

    public Student(String nameIn) {
        name = nameIn;
    }

    public String getName() {
        return name;
    }

    public void setName(String nameIn) {
        name = nameIn;
    }
}
```

Part 2 — Using a Class

```java
public class StudentDriver {
    public static void main(String[] args) {
        Student testStudent;
        testStudent = new Student("put your name here");
        System.out.println("Student name is " + testStudent.getName());

        // Use the mutator to change the student's name
        testStudent.setName("new name here");

        // Print the updated name
        System.out.println("Updated student name is " +
testStudent.getName());
    }
}
```

Part 1 — Implement class

```java
public class Product {
    private String name;
    private double price;

    public Product(String nameIn, double priceIn) {
        this.name = nameIn;
        this.price = priceIn;
    }

    public String getName() {
        return name;
    }

    public void setName(String nameIn) {
        this.name = nameIn;
    }

    public double getPrice() {
        return price;
    }

    public void setPrice(double priceIn) {
        this.price = priceIn;
    }
}
```

Part 2 — Create a Driver

```java
public class ProductDriver {
    public static void main(String[] args) {
        Product myProduct = new Product("Bread", 2.50);
        System.out.println(myProduct.getName() + " costs $" +
String.format("%.2f", myProduct.getPrice()));

        myProduct.setName("Lowfat Bread");
        myProduct.setPrice(3.00);

        System.out.println(myProduct.getName() + " costs $" +
String.format("%.2f", myProduct.getPrice()));
    }
}
```

Part 1 — Implementing a Program

```java
import java.util.Scanner;

public class DomesticLighting {

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

        System.out.print("Enter the price of the globe: ");
        double price = scanner.nextDouble();

        System.out.print("Enter the wattage of the globe (25, 40, 60, 75,
100): ");
        int wattage = scanner.nextInt();

        System.out.print("Enter the cost per kilowatt of electricity: ");
        double costPerKilowatt = scanner.nextDouble();

        double hours = 0;

        switch (wattage) {
            case 25:
                hours = 2500;
                break;
            case 40:
                hours = 1000;
                break;
            case 60:
                hours = 1000;
                break;
            case 75:
                hours = 750;
                break;
            case 100:
                hours = 750;
                break;
            default:
                System.out.println("Invalid wattage value.");
                return;
        }

        double totalCost = (wattage / 1000.0) * costPerKilowatt * hours; //
Convert wattage to kilowatt
        double costPerHour = totalCost / hours;

        System.out.println("Total cost of using the globe for its entire
lifespan: $" + String.format("%.2f", totalCost));
        System.out.println("Cost per hour of using the globe: $" +
String.format("%.2f", costPerHour));
    }
}
```