Creating a Class, Creating an Object, and Creating Multiple Objects in Java

Project Abstract

The purpose of this project is to demonstrate the creation of a class, the creation of objects from that class, and the creation of multiple objects in Java. A class in Java is a blueprint for objects, and objects are instances of a class. This project focuses on:

1. Define a class with properties and methods.
2. Create objects from the class and initialize their properties.
3. Handle multiple objects from a single class and manipulate them.

Tasks Overview

Task 1: Create a Class

Objective: Create a class called Car that has attributes and a method to display the details of a car.

Detailed Description: In this task, you are required to define a class named Car. The class will have two properties: model (String) and year (int). Additionally, it will have a constructor to initialize these properties and a method displayDetails() to display the car's model and year.

- Steps:

1. Define a class named Car.
2. Define properties: model and year.
3. Define a constructor that accepts model and year as arguments and initializes the class properties.
4. Define a displayDetails() method to print the car's model and year. in the below format:

Note: In the `displayDetails()` method, you need to print in the below format:

```
Car Model: <model>
Car Year: <year>
```

Task 2: Create an Object

Objective: Create an object from the Car class and display its details using the displayDetails() method.

Detailed Description: In this task, you will create an object of the Car class and display its properties using the displayDetails() method. The Car object will be initialized with specific values for model and year.

- Steps:

5. Instantiate an object car1 of the Car class using the new keyword.
6. Pass values to the constructor for model and year (e.g., "Toyota" and 2020).
7. Call the displayDetails() method on car1 to display its properties.

Task 3: Create Multiple Objects

Objective: Create multiple objects from the Car class and display their details.

Detailed Description: In this task, you will create multiple objects from the Car class and display their details using the displayDetails() method. Each object will have different values for model and year.

- Steps:

8. Instantiate additional objects car2 and car3 of the Car class with different values for model and year (e.g., "Honda" and 2018, "Ford" and 2022).
9. Call the displayDetails() method for each object to display their respective details.

Execution Steps to Follow:

1. **All actions like build, compile, running application, running test cases will be through Command Terminal.**
2. **To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) Terminal New Terminal.**
3. **This editor Auto Saves the code.**
4. **If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.**
5. **These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.**
6. **To run your project use command:**

```
mvn compile exec:java
-Dexec.mainClass="com.yaksha.assignment.ClassCreationAssignment"
```

7. **To test your project test cases, use the command**

```
mvn test
```

8. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**