ORACLE
Academy

Java Fundamentals
7-3: The Static Modifier and Nested Classes

Practice Activities

Lesson Objectives:

- Create static variables
- Use static variables
- Create static methods
- Use static methods
- Create static classes
- Use static classes

Vocabulary:

Identify the vocabulary word for each definition below.

	Is a method that is available for use without first creating an instance of the class. It is declared by preceding its definition with the static modifier.
	Is any class implemented as a nested class within another class. By definition, all inner classes are members of the container class by composition.
	Any Java class-level variable that is declared with the static modifier. This means only one instance of the class variable can exist in the JVM regardless of the number of class instances.
	Is a keyword that makes a variable, method, or inner class available without first creating an instance of a variable.
	Is an inner class. Inner classes are defined within a parent or container class and are members of the container class by composition. In fact, inner classes are the only way you can create class instances through composition.
	Is an inner class that is available for use without first creating an instance of the container class. It is declared by preceding its definition with the static modifier.
	Any Java method defined with a static modifier. It is accessible outside the class when a public, protected, or default access specifier precedes it. It is private and inaccessible outside of the class when a private specifier precedes it. Class methods are available without first creating an instance of the class.
	Is a variable that may be available outside of a class without first creating an instance of a class. It is declared by preceding the variable name with the static modifier.

Try It/Solve It:

1. Create classes that will be used to compare the use of static and instance variables.
   a. Create a new package named “vehicles”.
   b. Create and implement the following concrete class:
      i. Create a class named “Vehicle”.
      ii. Create your static variables as follows:
         a. A public static String variable named “MAKE” with a value of “Augur”.
         b. A public static int variable named “numVehicles” with an initial value of 0.
      iii. Create your instance variables as follows:
         a. A private String variable named “ChassisNo”.
         b. A private String variable named “model”.
      iv. Create a constructor that takes a single formal parameter of type String named “model”. The constructor should execute the following tasks:
         a. Increment the value of the static variable “numVehicles” by one.
         b. Set the value of the instance variable “chassisNo” equal to the concatenation of “ch” with the value held by “numVehicles”.
         c. Set the instance variable “model” equal to the value of the parameter “model” (hint: you will have to use this here).
         d. Have the constructor display a message that states “Vehicle manufactured”.
      v. Implement two pairs of getter and setter methods that will allow you to get and set the values of the two instance variables.
   c. Create and implement the following driver class:
      i. Create a class named “TestVehicle”.
      ii. Create a static main method that tests the following:
         a. Using the value of the static variable “MAKE” create the following output message:
            - “Manufacturer: Augur”
         b. Using the value of the static variable “numVehicles” create the following output message:
            - “Number of vehicles manufactured: “
      iii. Run your program! You will see that the values that are identified as static are created at runtime and can therefore be accessed.
      iv. In your main method use the “chassisNo” variable to ” create the following output message:
            - “The chassis number is *chassisNo*”.
      v. Run your program! You will see that the program will not run. This is because we haven’t yet created an instance (object) of the Vehicle class.
      vi. In your main method create a vehicle object named vehicle1 above the output statement for the chassis number.
      vii. Update the chassis number output statement to use dot notation to identify which vehicle’s chassis we want to see (use vehicle1).
      viii. Run your program! You will see that the program now shows both the static and the instance variables.
      ix. Update the existing code to also show the model of the car.
      x. Create a second vehicle (named “vehicle2”, use “Edict” as the parameter for the constructor) and display the instance variables of that object.
      xi. Create a toString() method in the Vehicle class that will display the vehicle’s make, model and chassis number to screen using a different line for each output statement.
      xii. Display the contents of the vehicle in your main method by using the toString() method.

xiii. Add a final output method that will display the total number of cars manufactured.

The output of the program should look like this:

```
Manufacturer: Augur
Number of vehicles manufactured: 0
Vehicle manufactured
The vehicle is manufactured by: Augur
The model type is Vision
The chassis number is ch1
Vehicle manufactured
The vehicle is manufactured by: Augur
The model type is Edict
The chassis number is ch2
Number of vehicles manufactured: 2
```

2. To highlight the fact that static variables are stored in a single memory space that is accessed by each instance of a class we are going to amend the code created in question 1.
    a. In the main method, above the line that displays the total number of cars manufactured add the following line of code:
        - vehicle2.setMake("Seer");

       We are using vehicle2 to amend the value but we could use any instance name.
    b. Add two output statements under this line that will use the toString method to display the value of vehicle1 and vehicle2.

The output of the program should look like this:

```
..
..
..
The vehicle is manufactured by: Seer
The model type is Vision
The chassis number is ch1
The vehicle is manufactured by: Seer
The model type is Edict
The chassis number is ch2
Number of vehicles manufactured: 2
```

3. Using the solution created in question 2 you are going to create a nested static class that will hold the details of the engine used within the vehicle.
    a. Open the vehicle class.
    b. Under the constructor method create a public static class named Engine.
    c. Create two private static final variables named MAKE and CAPACITY. Make will hold text while capacity will store whole numbers.
    d. The values that you should assign to the variables are “Predicter” and “1600”

e. Do not create a constructor for this class.

f. Create two public static getters for both of the variables created.

g. Update the toString method in the Vehicle class to display the engine make and model. Remember the Engine is a static class.

<u>The output of the program for each vehicle should now look like this:</u>

```
The vehicle is manufactured by: Seer
The model type is Edict
The chassis number is ch2
The engine make is Predicter
The engine capacity is 1600cc
```

4. You are going to adapt your code to use an inner static class that can return instance information from its container class.

   a. Change your Engine declaration to match the following:
      - public static class Engine extends Vehicle {

   b. Create a constructor that accepts a String parameter named model.

   c. The constructor should have a single instruction that send the model variable to the super constructor.

   d. Go to your main method and create an Engine object named vehicle3 above the line of code that displays the total number of cars manufactured. To do this you will need to follow these guidelines:
      - Outerclass.InnerClass object name = new Outerclass.InnerClass (parameter);
      - Vehicle.Engine vehicle3 = new Vehicle.Engine("Fortune");

   e. This gives you access to both the methods and fields of the inner class as well any methods and fields in its enclosing class. Create an output statement that will use the appropriate getter methods to display the following:
      - “Vehicle number ch3 is a Fortune model and has an engine capacity of 1600cc”

   f. To see the difference between the different types of vehicles try to create the previous statement using vehicle1 or vehicle2 (it won’t work as they don’t have access to the inner workings of the static Engine class).