Bridge

In this lab, you will create an example of a Bridge application.

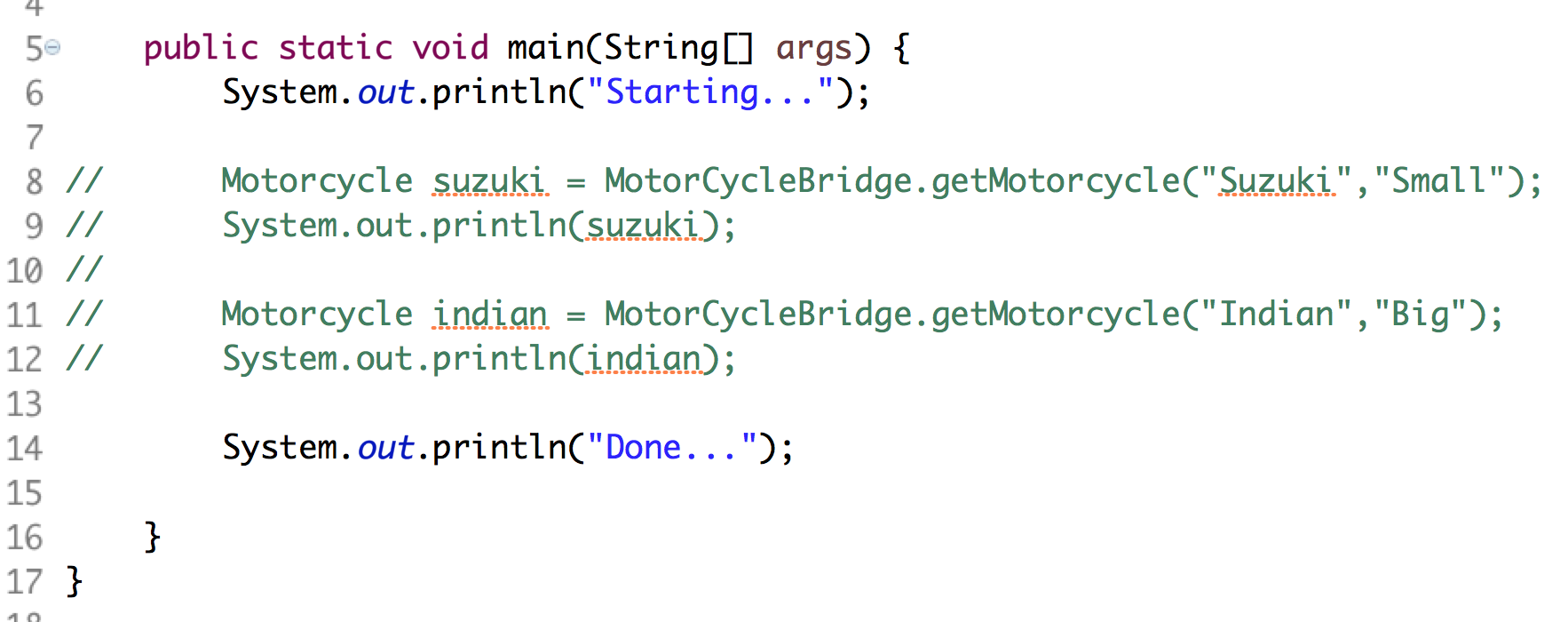
# Objectives

In this lab, you will

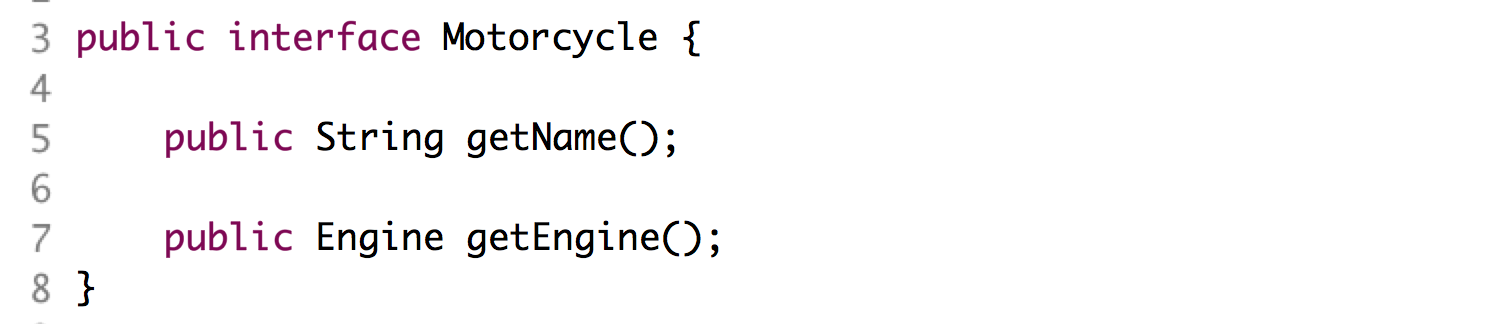
* Create a Factory with two Proxies
* The Factory gets the first proxy, then passes that proxy to the second
* It returns the second proxy

# Exercise

1. The Bridge design pattern represents the usage of two or more Proxies. The client calls a Factory to get one Object then uses that Object as a parameter to another Factory to get the final Object. In this exercise, the client wants to create one of several Motorcycle objects with one of several Engine sizes.
2. In Eclipse, in the exercises project, open the package com.paypal.patterns.Bridge to view the project files.
3. Open Tester.java shown below:

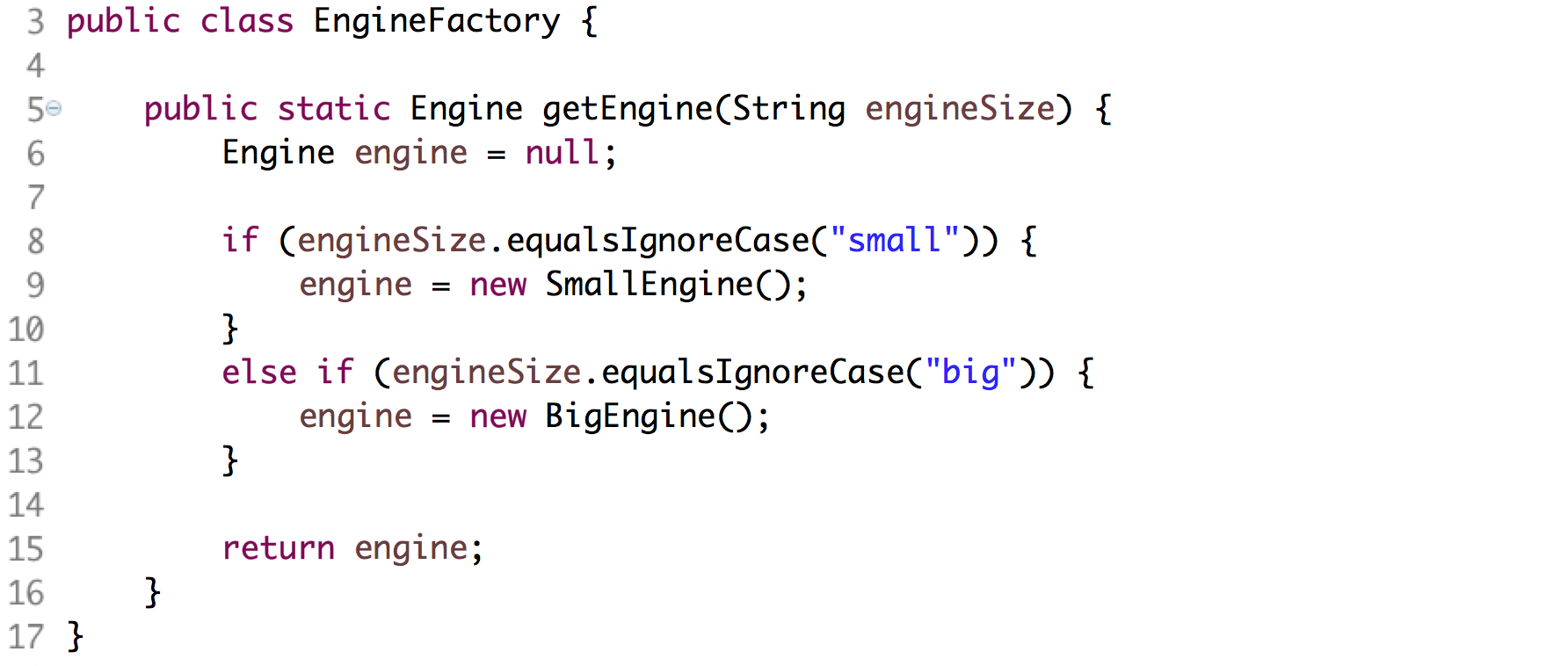


1. In the above, Tester calls the Bridge to get two different motorcycles. The MotorCycleBridge is responsible for converting the strings to objects and returning the correct one.
2. Examine the simple Motorcycle and Engine interfaces below:

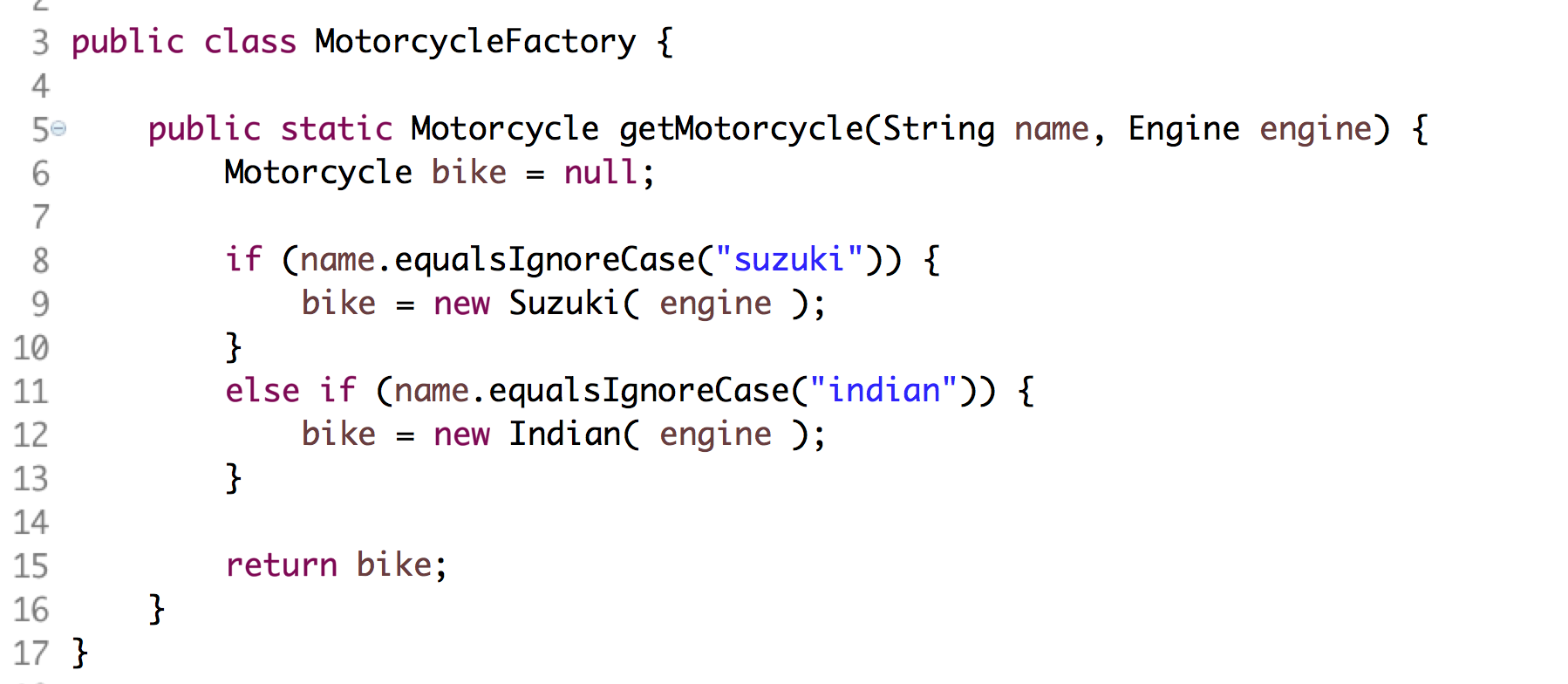




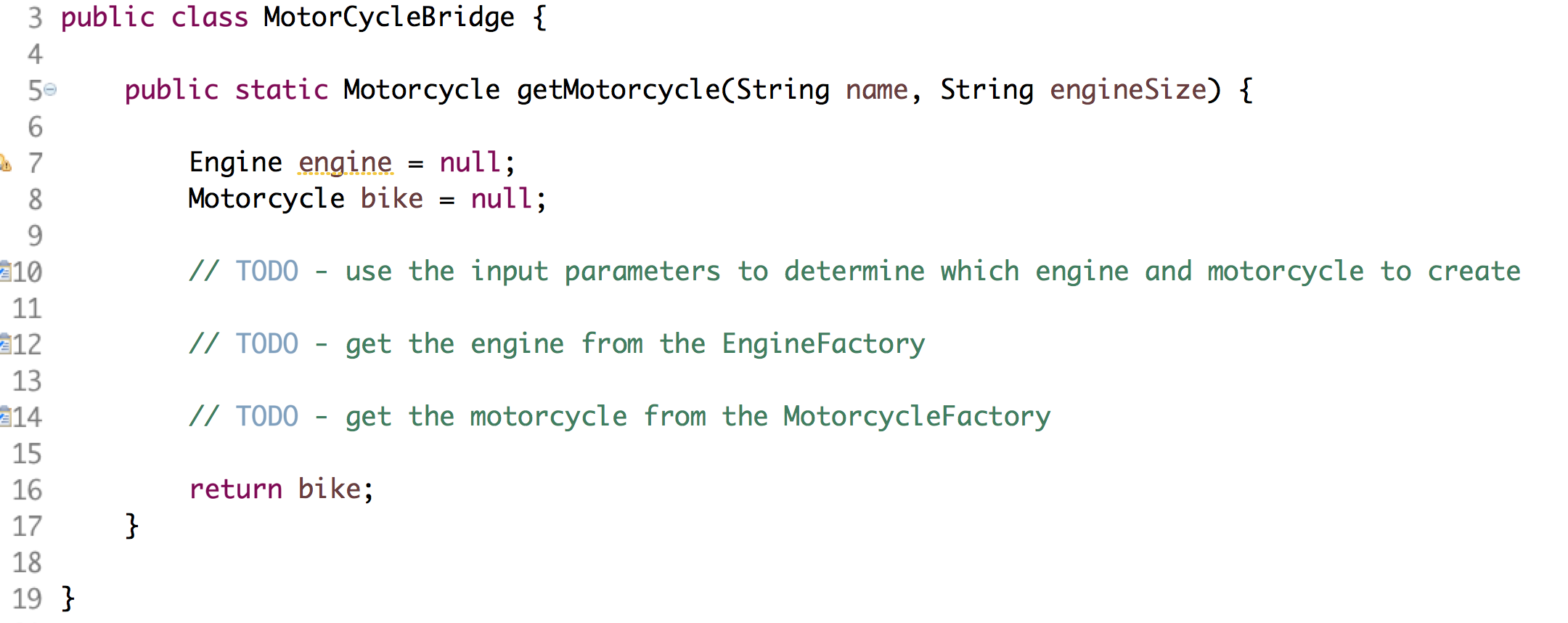
1. The realization of these interfaces reside in the classes: Suzuki, Indian, BigEngine, and SmallEngine in the same folder. These classes implement simple Java beans.
2. Examine the EngineFactory:



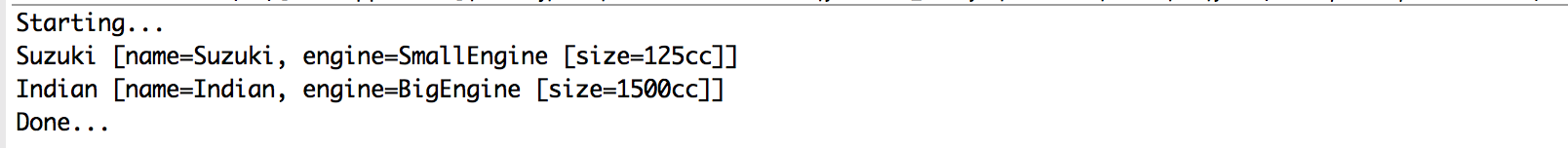
1. In the above, the EngineFactory creates an engine based on the input string, engineSize.
2. Examine MotorcycleFactory:



1. In the above, notice one of the input parameters is an Engine, created by the EngineFactory earlier.
2. We could have created a Motorcycle by passing strings to the getMotorcycle() method… But, then, the Motorcycle is tightly coupled to the Engine. The getMotorcycle() would have to know how to create an Engine.
3. The Bridge pattern is the connection between these two factories. It allows the two factories to be loosely coupled.
4. Edit MotorCycleBridge.java as shown below:



1. In the TODOs, create calls to the two factories.
2. Uncomment the lines in Tester.java and execute the file as a Java application. You should see:



Congratulations. You have completed this lab.