Fall 2020

CS149 – Introduction to Programming

Programming Assignment 5

Testing and Debugging the `Converter` Class

Overview

The project manager at *DukeDash* hired a University of Virginia graduate to add some code to the original `Converter` class (i.e., the version from programming assignment 2). However, the project manager is concerned about the quality/correctness of the code and wants you to test it and debug it if necessary.

Specifications Provided to the Other Programmer

The other programmer was provided with the following specifications for the `Converter` class:

1. The `Converter` class must have the following additional methods (over and above those from programming assignment 2):

   ```
   private static double inchesToFeet(double inches)
   private static double inchesToMiles(double inches)
   private static double mphToRPM(double mph)
   ```

   The purpose of each method should be apparent from its name.

2. The `mphToRPM()` method must convert from miles per hour (mph) to rotations per minute (RPM). It must assume that the *diameter* of the tire/wheel is 27 inches.

Existing Tests

A JMU graduate who works at *DukeDash* created a test suite for the **original** version of the `Converter` class. It is named `ConverterTest.java` and, as it should be, is a main class. (Note: It uses the `JMUConsole` class that you have used in the past and the `Test` class from the lab on test harnesses.)

Your Tasks

You are responsible for writing a test suite for these new methods and testing the existing implementation. If necessary, you must correct any defects in the code.

Recommended Process

1. Read and understand the entire assignment.
2. Create a directory/folder (e.g., named `pa5`) that will hold all of the files for this assignment.
3. Copy `JMUConsole.class` and `Test.class` into the directory you created for this assignment. (You should have `JMUConsole.class` in the directory you created for earlier programming assignments. You should have `Test.class` in the directory you created for the lab on test `Test.class` harnesses.)
4. Download the .zip file containing the source code for the questionable implementation of the `Converter` class and the `ConverterTest` class and **unzip it into the directory you created for this assignment**.
5. Read and understand this version of the `Converter` class. (**Do not use your version!**)
6. Read and understand the `ConverterTest` class.
7. Compile the `Converter` class and the `ConverterTest` class. They should not contain any syntax errors.
8. Check the style of the `Converter` class and the `ConverterTest` class. They may contain style errors; if they do, fix them.
9. Execute `ConverterTest` so that you know the code from the **original** `Converter` class is correct.
10. Add a method named `inchesToFeetTest()` to the `ConverterTest` class. This method should have enough tests to ensure the correctness of the `inchesToFeet()` method in the `Converter` class.
11. Add an invocation of `inchesToFeetTest()` to the `main()` method of the `ConverterTest` class. (Note: Don't eliminate any other tests. You want to make sure that you don't break anything that was working!)
12. Execute the test suite.
13. Debug the `inchesToFeet()` method in the `Converter` class if necessary.
14. Repeat steps 10-13 for both the `inchesToMiles()` method and the `mphToRPM()` method.

15. Submit both the corrected version of `Converter.java` and your implementation of `ConverterTest.java` in a file named `pa5.zip` using Autolab. Do not include any other files in the `.zip` file.

Grading

Your grade on this assignment will depend both on the quality of the tests and the quality of the corrected `Converter` class.

Your code will first be graded by Autolab and then by the Professor. The grade you receive from Autolab is the maximum grade that you can receive on the assignment.

Autolab Grading

Your code must compile (in Autolab, this will be indicated in the section on “Does your code compile?”) and all class names and method signatures comply with the specifications (in Autolab, this will be indicated in the section on “Do your class names, method signatures, etc. comply with the specifications?”) for you to receive any points on this assignment.

Autolab will then grade your submission as follows:

Conformance to the Course Style Guide:	**20 points** (Partial Credit Possible)
Coverage/Quality of Your Tests:	3**0 points** (Partial Credit Possible)
Correctness:	5**0 points** (Partial Credit Possible)

Autolab will only provide limited hints because, at this point in the semester, you should not be using Autolab to test and debug your code, you should be doing it yourself. For the coverage/quality of your tests it will indicate which methods are not being adequately tested, but will not provide any details. For the correctness of the debugged code, it will provide very little information.

Manual Grading

After the due date, the Professor may manually review your code. At this time, points may be deducted for inelegant code, inappropriate variable names, bad comments, etc.