**Workout 2020-09-04 – Decorators Names: \_\_Sydney O’Dell\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructions:** Put your answers to the first set of questions in this document. Make sure your name is at the top. Create a project for your code specified in Problem 2. When you are finished, zip up the workout folder and submit it to this assignment on ASULearn.

**Questions**

1. Define the following terms in your own words.
   1. Extension (from the Open-Closed Principle)
      1. Code or a file that opens a working program up to extra features and abilities.
   2. Wrap an object
      1. Wrapping an object changes the format of data to the same format as the rest of the code.
   3. Decorator
      1. Decorators have multiple smaller classes that act as additions to a Concrete Component. They can be added in different orders and keep the code as simple as possible.
   4. ConcreteComponent
      1. A simple object that gets added to by the decorators dynamically.
   5. Abstract Decorator
      1. An extension that holds all the decorators and is wrapped
   6. FilterInputStream
      1. The component that is being decorated.
2. Modify the textbook’s InputStream program so that, in addition to converting all the alphabetic characters to lower case, add two other decorators:
3. Count the total number of characters read from the InputStream.
4. Filter out all white space characters.

The output (System.out) of your program should be all characters in the file displayed without white space characters followed by a blank line and the text:

Total Characters Read: xxx

Keep the input file name the same as in the book's code. Your count should be the count of non-whitespace characters only (NOTE: This specification does NOT change your decorator!)

You can access the textbook’s code at <https://github.com/bethrobson/Head-First-Design-Patterns/tree/master/src/headfirst/designpatterns/decorator/io>.