CST8116 Assignment 02 (22F) use of static

There are two classes in the starter code for Assignment 02, that have every class member marked static, both also have a private constructor. This may be causing confusion for some students.

# Overview

Generally speaking, when designing a class, we need to decide if:

* the class will be used for objects that need to have some sort of independent internal state (state is variables with data, at a given point in time or program execution), these objects act as a model for something.
* or the class simply contains utility methods or data that we will use but never need to create objects from.

For example:

If I create Person objects, based on a class Person, each can have its own name, age, phone-number and so on.  
In other words very little if anything in the class will be static, everything will be instance variables and methods, with typical constructors. Each object models a separate person.

If I need to perform common math functions, as well saw in week 2 of the course, we can use a utility class like Math.  
In other words everything in class Math is static, and we do not to construct any objects to use the functionality.

Note that class Math has a private constructor, this prevents other programmers from doing something like:  
Math myMath = new Math(); // does not work.

We are forced to use class Math as a class, e.g. Math.PI, Math.pow(a,b), Math.sqrt(a) and so on.

# Assignment 02 Classes and Class Roles

## Class User

* Utility class, all methods should be static, and there is a private constructor.
* Instead of User user = new User(); treat this class similar to class Math and just use it via the class name

int number = **User**.inputPositiveInteger("Enter diameter");

* The starter code had an omission, and all methods in this class should be static.

## Class CircularSawBlade

* This class models a real-world object, most class members are not marked static
* There are however some class-level constants marked public static final which can be used elsewhere in the program.

## Class CircularSawBladeVerifier

* Utility class, all methods should be static, also has private constructor
* Simply use the class e.g., boolean result = **CircularSawBladeVerifier**.isDiameterInTolerance(myBlade);
* Note that the methods to be worked on need to access constants from class CircularSawBlade, no need to make CircularSawBlade objects for these, simply use class CircularSawBlade similar to using class Math for PI

double difference = Math.abs( blade.getDiameter() – **CircularSawBlade**.EXPECTED\_DIAMETER );

// blade is an object with separate data for each saw blade, access the constants via the class name however

## Class Assignment02

* This is a third type of class, typically for Java programming, the only job or role for this type of class is to have a main method to start the application running.