COMP 2526 PROGRAMMING CHECKLIST

1. Your code must compile.
2. Your code must not generate any compiler warnings (the little yellow flags in the left-hand margin).
3. Your code must execute without crashing.
4. Your code must not generate any Checkstyle complaints.
5. All classes require Javadoc comments including @author tag AND @version tag. Class comments go after the package and import statements. There should be no blank lines between a class comment and the class it describes.
6. Public constants require Javadoc comments, private constants do not. Constants should be the first thing in your class.
7. Constants should be static and final, are often public, and should be followed by instance variables. Instance variables have private or protected visibility, never public or default visibility.
8. Public and protected methods require Javadoc comments including @param tag(s) and @return tag where needed.
9. Method comments must begin with verbs describing what they do, i.e., Calculates, Returns, Sets, Prints, etc. The @return and @param tags go AFTER the description.
10. Private methods require non-Javadoc comments (the green comments).
11. Do comment complicated logical blocks of code inside methods with sparse, clear inline comments.
12. Do not use magic numbers (you must use constants instead).
13. All method parameters that are object references must be made final (so we don't forget parameters are passed by value)
    1. Nice to prevent erroneous assignments, and necessary if parameter is referenced by inner class, but that is perhaps a little advanced.
    2. References made final mean that the reference, once pointing to an object, cannot be changed to point at a different object.
    3. Final arguments prevent accidental reassignment of an object inside a method.
    4. Final **methods** prevent subclasses (those that inherit the method) from changing its meaning.
    5. Final **methods** are more efficient (the methods become inline, thus avoiding the stack and generating overhead).

14. Inheritance is used for is-a relationships only.
15. Common elements in subclasses are pushed up the hierarchy to the superclass (when logical to do so).
16. Data and methods that work together are encapsulated in the same class.
17. Code duplication must be minimized.
18. Default values for primitives are set in methods.
19. Don't squeeze your code together. Put a blank space on either side of the equal sign, for example. I will be looking at the readability and clarity of your code.
20. Document any variations from assignment requirements in a readme file.
21. All of your program classes must be in package ca.bcit.comp2526.xxx (replace xxx as required by the assignment).
22. Constructors: The first line of every constructor must be either
    1. A this call to another constructor in the same class or
    2. A super call to a parent constructor.
23. In general, we enforce a maximum **method** length of 20 lines of code. Aim for 10 (excluding braces).