

Coding Standards

Coding standards are a set of mandatory formatting rules that govern the production of quality, legible code. Coding in a uniform style makes it easier for you to read code from your instructor and classmates. If you already have programming experience, you may have gotten used to different standards. Like any new development shop you work in, please comply with these standards for all code. Your suggestions for improvement are welcome but until the coding standards are updated, code will be graded for compliance with the existing standards.

1. Variable names should be descriptive. Simple variables such as a, t or x should be avoided. Variable names should begin with a lowercase letter and the first letter of each word should be capitalized. For example, firstPerson. The use of fp here would not be a descriptive variable name.
2. Class names should begin with an uppercase letter and the first letter of each word should be capitalized. For example, GoodClassName.
3. Constant names should be all upper case. Words should be separated by and underscore. For example, CONSTANT_NUMBER.
4. Indentation should provide the reader with a context for their location. Brackets should start on the same line and the start of this line should line up with the matching closing bracket.
5. Comments will be used. Poorly commented code will receive a lower grade. Comments should include descriptive text that will ensure a reader understands the code. All variables should be commented to ensure there is no confusion about the variables purpose.
6. Blank lines should be used after comments, variable declaration blocks, constructors and after each method.
7. Javadoc comments should be used to describe classes and methods.

- a. Example class comment

```
/**Class: ClassName
 * @author Your Name
 * @version 1.0
 * Course: ITEC XXXX Fall 2019
 * Written: August 18, 2019
 *
 * This class - now describe what the class does
 */
```

- ClassName should be the name of the class. It should not be ITEC 2140, ITEC 2150 or some course number. This is used to identify you class. If it is a public class, it will be the file name without the .java extension.
- Author - You must include your name in your program. Programs submitted without a name will receive a deduction
- Version - should start 1.0. For minor changes should receive a point upgrade, 1.1 etc. If a second submission is required, this should be updated to 2.0 etc.
- Course - what course and semester is this program for.
- Written - what date did you create/modify this program.
- Class Description - describe what the class should do at a high level.

b. Example Method comment

```
/** Method: Method Name
 * Convert calendar date into Julian day.
 * Note: This algorithm is from Press et al., Numerical Recipes
 * in C, 2nd ed., Cambridge University Press, 1992
 * @param day day of the date to be converted
 * @param month month of the date to be converted
 * @param year year of the date to be converted
 * @return the Julian day number that begins at noon of the
 * given calendar date.
 */
```

- Method – Must be a descriptive name of the method. Helps programmers reading the code understand the purpose of the method. `methodName()` is not a good name. `printResults()` is a better name. Describes what the method is doing.
 - @param – This is a Javadoc comment describing each argument to a method. Notice you can have multiple param lines, one for each argument.
 - @return – Describes the results of the method and the type of the return value.
8. Variables should not be defined at the top of a code block but rather define just prior to use. Keep in mind scope rules may affect definition locations.
 9. In an if statement or a loop, use brackets ({}) to delimit the code that you want to execute even if you only have one statement in the if or else block.
 10. Use blank space to improve readability. For example, when writing an expression with binary operators, use a blank around the operators.
Good – $(-b + 3)$
Not good – $(-b+3)$
 11. Methods should have at most 30 lines of code.
 12. Main methods should appear at the end of a source file. If not at the end, it should be the first method in the class.