**Programing Style and Documentation**

**CS200**

***Syntax vs Style***

Every programmer will develop their own “style” of programming. This is both a blessing and a curse. Programmer style allows for creativity in problem solving, but it can also cause problems when working with others. This course will differentiate between syntax and style. Even though code may be syntactically correct, it must also follow certain rules of style. While students can use any programming method to solve problems, their code must follow the rules of style listed below to be fully correct.

***File Documentation: Block Documentation***

Every program file submitted must have block documentation at the start of the file. This summarizes information about the overall program and contains your name and the name of the assignment. Any file without documentation will not receive full credit.

***File Documentation: Citing References***

If any external code is used, it must be properly documented within the CREDIT sections shown below. If any external code is used and not documented, it is considered plagiarism.

***File Documentation: Method Documentation***

Every single method needs its own block documentation. It is a different format than the file documentation, and different information is necessary. The template for method documentation is as follows:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* (Method Name) \*

\* \*

\* Purpose: (briefly describe the purpose) \*

\* Parameters: (list all parameters and their purpose to the \*

\* method) \*

\* Return Value: (Describe the value returned, if any) \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

***Naming Conventions:***

1. Variable Names

Use meaningful variable names to illustrate what is ?ccc?. For multiple word variable names, use cameCaseNames, or separate\_words\_with\_underscores. Either naming convention is fine but use it consistently in your programming. (TA’s note, Java convention is to use camelCasing, different languages have different conventions.)

1. Class Names

Start all class definitions with a capital letter, such as class *Utilities* not *utilities*. Do not capitalize variable names, as capitalized names are reserved for classes.

1. Parameter Names

Oftentimes, parameters are immediately stored in attributes, and then never used again. If a parameter name and purpose match an attribute, use the same name, but end with an underscore “ \_ “ . Ex) attribute *value* –> parameter *value\_ .*  Consequently, if a parameter is not used for this purpose, do not use the underscore.

1. Indentation and code blocks

In programming, visual organization is important. Always use indentation of 4 spaces per level, and make sure it is consistently used. When using “brackets” (curly braces {}) for code blocks, do so as shown below:

|  |  |
| --- | --- |
| Do | Do Not |
| public void xxxxx  {  if(xxxx)  {  xxxxx  }  else  {  xxxxx    } | public void xxxxx{  if(xxxx){  xxxxx}  else{xxxxx}  } |

1. Use of “this.”

Java has a *this.* (this dot) command that allows reference of attributes or methods. In Java the use of “this.” Is syntactically optional, in that it is obscured. For this class, be specific and use “this.” When referring to any attribute. It is more readable and illustrates the point that it is referring to an attribute or method of that object.

1. Complete block symbols

It may seem tedious to extend the column of stars on the right side. However, this makes the documentation easier to see, from a big picture perspective. In objects with many methods, that column is quite helpful for quickly determining code vs documentation.

***File Documentation:***

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* (title of assignment) \*

\* \*

\* PROGRAMMER: (Your Name) \*

\* CLASS: CS102 \*

\* ASSIGNMENT: Assignment # \*

\* INSTRUCTOR: Dean Zeller \*

\* TA: Robert Carver \*

\* SUBMISSION DATE: (date of submission) \*

\* \*

\* DESCRIPTION: \*

\* (Describe the program implemented. Feel free to use material\* \* from the assignment writeup.) \*

\* \*

\* CREDIT: \*

\* Source #1: \*

\* Program: (program name) \*

\* Programmer: (programmer name) \*

\* Location: (where original code was found) \*

\* Modifications: (list the modifications made to the \* \* original code) \*

\* \*

\* COPYRIGHT: \*

\* This program are © 2018(your name) and Dean Zeller. This is \* \* original work, without use of outside sources. \* \* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/