|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Artifact Information** | | | | | | | | |
| **Artifact ID** | | **Artifact Title** | | | | | | |
| SCS-001 | | Software Coding Standard | | | | | | |
| **Capstone Team** | | | | **Revision** | | **Artifact Date** | | |
| Capstone Team 27 - Granustem | | | | 1.0 | | Jan 11, 2019 | | |
| **Prepared by** | | | | **Checked by** | | | | |
| Ben Alexander | | | | Jonathan Meldrum | | | | |
| **Revision History** | | | | | | | | |
| **Revision #** | **Date** | | **Prepared by** | | **Checked by** | | **Description** | **Approved by** |
| 1.0 | Jan 11, 2019 | | Ben Alexander | | Jonathan Meldrum | | Initial Version | Reese Bastian |

1. Purpose

The purpose of this artifact is to outline the coding standard that will be used within the software subsystem. It follows the PEP-8 Style Guide for Python very closely.

2. Coding Standard

1. Follow PEP-8 naming conventions. See chart at the end of this file
2. Only loops can have single-letter variables or names without good reason
3. Surround top-level functions and classes with two blank lines
4. Surround method definitions inside classes with a single blank line
5. No lines longer than 80 characters
6. If a function declaration requires multiple lines, use the following syntax:

def function(

arg\_one, arg\_two, # 8 spaces before code

arg\_three, arg\_four): # 8 spaces before code

return arg\_one # 4 spaces before code

1. Long arithmetic operations should use the following syntax:

total = (first\_variable

+ second\_variable # 8 spaces before code

- third\_variable) # 8 spaces before code

1. Indentations should be made with 4 consecutive spaces, rather than a Tab
2. Lists with opening and closing braces/brackets should use the following syntax:

list\_of\_numbers = [

1, 2, 3, # 4 spaces before code

4, 5, 6, # 4 spaces before code

7, 8, 9 # 4 spaces before code

] # 0 spaces before closing brace

1. Comments should have proper grammar, capitalization, and punctuation. They should typically be full sentences, but this isn't always necessary
2. Block comments (describing code) and Docustrings (describing functions, represented with """ to open and close the comments) should be the first lines just within the nested code it describes. They should be indented to the same level as the code they describe
3. Inline comments must be preceded by a blank space between the end of the code and the #. There should also be a space after the #
4. Surround binary operators with a single space on either side except for the following exceptions:
   1. When = is used to assign a default value to a function argument, do not surround it with spaces
   2. When there’s more than one operator in a statement, only add whitespace around the operators with the lowest priority, especially when performing mathematical manipulation. This can also be applied when there are multiple conditions in an if statement
5. No trailing whitespace
6. A single space between comma separated items in a list, but no spaces at the beginning nor the end of the list between the brackets
7. Don’t compare Boolean values to True or False using the equivalence operator
8. Frequently use the pycodestlye Linter to see if any uncaught changes should be found
9. All files should be UTF-8 encoded
10. All imports should be on separate lines, unless you are importing multiple sub-imports from a single importing
11. Imports should be grouped in the following order, with a blank line between each group:
    1. Standard library imports.
    2. Related third party imports.
    3. Local application/library specific imports.

