**CODING GUIDELINES DOC (based on PEP8):**

**Naming Conventions**

**Package and Module Names**

* Packages and modules should have short, all-lowercase names.

**Class Names**

* Class names should be CapitalizedWords (or CapWords, or CamelCase, or StudlyCaps) to improve readability and be easily distinguishable from variable and function names.

**Function and Variable Names**

* Function names should be mixedCase for readability and internal consistency.
* Variable names follow the same convention as function names.

**Method Names and Instance Variables**

* Follow the function naming rule: mixedCase.
* Use two leading underscores for non-public methods and instance variables so they will be mangled by Python and be inaccessible by name when modules are imported.

**Constants**

* Constants will be defined in module and written in all capital letters with underscores separating words

Example: MAX\_NUMBER

**Indentation**

* Use a single tab per indentation level

**Layout**

**Whitespace**

* Always surround these binary operators with a single space on either side: assignment (=), augmented assignment (+=, -= etc.), comparisons (==, <, >, !=, <>, <=, >=, in, not in, is, is not), Booleans (and, or, not).

**Exception Handling / Logging**

What are your standards for exception handling & logging, is it all home grown or are you using a third party tool? How should it be used?

**Commenting**

* Block comments should be added to each method and start with # and a single space
* Inline comments should be placed on a separate line above the line of code being commented on.
* Documentation strings are written for all public modules, functions, classes, and methods. DocStrings should start with “““ and end with ””” on the line following the end of the comment.

For example:

"""

Return a foobang

Description of method pre/post condition & assumptions

"""

* For one liner docstrings, please keep the closing """ on the same line.

**Exposure**

* All instance variables should be inaccessible by name once and instance of object is created
* Instance variables can be accessed through get methods.

For example: getDescription()