**Coding style guidlines**

**Indentation and scope define:**

* One must always use tab-based indentation.
* Each scop inner scop should be separated by one tab from the outer scop, avoid over indenting.
* Opening bracket should be on the same row as the scop declaration, closing bracket should be on the same column of the declaration.
  + Exception: one line scop should have no brackets.
* maximum line length – 120 characters .

Example:

Void myFunc(List<int> numbers){

For (int number : numbers){

If (number >= 10) System.out.println(number)

}

}

**spacing and blank lines:**

* Methods and Class declarations should be should be separated be a blank line.
* All identifiers are surrounded with whitespace.
  + Exceptions:
    - All declarations should be immediately followed by a left parenthesis.
    - All array dereferences should be immediately followed by a left square bracket.
    - The unary operator should be immediately preceded or followed by the operand.
    - The cast should be written with no spaces.
* One blank line to separate chunks of code is permitted for clarification.

**Class member ordering:**

1. Final Attributes
2. Attributes
3. Constructors
4. methods

**Parentheses:**

* Parentheses should be used in expressions not only to specify order of precedence,  
  but also to help simplify the expression. When in doubt, parenthesize.

**Identifiers and Naming conventions:**

* All class names should be written in PascalCase (e.g. MyClass)
* All “MAGIC NUMBERS” should be written in UPPER\_CASE (e.g. MAGIC\_NUMBER)
* All other identifiers including but not limited to method, variables and functions should be written in in camelCase (e.g. int myVariable)
  + Exception: acronyms should be all caps (e.g. int ID)
* All identifiers use letters ('A' through 'Z' and 'a' through 'z') and numbers  
  ('0' through '9') only. No underscores, dollar signs or non-ascii characters.
  + Exceptions: Private attributes should start with an underscore (e.g. this.\_privateAttribute)
* One letter variables are permitted only as iterators or according reasonable naming conventions

**Documentation:**

* Each identifier should be named a proper self-explanatory name that does not require and explaining comment.
* broadly, code should be written in a self-explanatory way that does not require explaining comments. Use comments to explain tricky parts of your code that require a few reads to understand
* Inline code comment should be written with //
* Any other comment or documentation should be written with /\*\*/
* Documentation blocks should have blank line before them and the code in the line below