- General
    - Classes
        - Model-view
        - No GOD class
    - Naming
        - Use camel case
        - Reasonable name
- Variables
    - Constant variables name must be in upper case
    - Name should make it easy to understand:
        - State what it is
        - How its used
    - Single letter variables are okay for iterators
- Classes
    - Naming
        - Camel case, starts with uppercase
    - Headers
        - Public members first, followed by private
        - Variables declared before methods
        - Slots declared before signals
        - Separate private from public slots
            - Public members
            - Private members
            - Public Slots
            - Private Slots
            - Signals
        - Declare method parameter names to avoid ambiguity
        - Include a header guard (ex. “#ifndef CLASS_NAME_H“...)
    - .cpp files
        - Include methods in the same order they are declared in header
    - Includes
        - Source files should include all necessary headers directly, not indirectly through other (or corresponding) header files.
        - Header files should include only the necessary headers for types needed in the class definition.
        - Use the angle bracket include format, where possible (ex. “#include <QTimer>” preferred vs. “#include “QTimer” ”)
- Methods
    - Camel case, starts with lowercase
    - Actionable name (“doSomething” preferred vs. “somethingToDo”)
    - Integrate nearly repetitive methods as helper method
- Indenting
    - 4-space indent per indent level
- Whitespace

    - Each operator should be preceded and followed by a space.
        - Ex: int a = 1 + 1;
    - Blank lines should be used to separate code into understandable chunks
- Comments
    - Doxygen format for methods are put in header files
        - @brief is required
        - @return is required
        - @param may be left blank if it is self explanatory with the @brief description
    - Header comments
    - Describe what does this function/variable do instead of the result of running