Week 2 Quiz 1

1. Consider the following prompt to an LLM, asking it to document a library of Python code. Which of the principles of documentation writing is MOST likely to be broken by the output generated?

   "As an expert Python developer, add docstrings to the methods in this library following the ReST style. Keep comments concise but write in an approachable style and avoid jargon as this library will often be used by newer teammates....[paste library code]..."

   - Be clear and concise
   - Think of your audience
   - [✓ Avoid redundancy]
   - Follow language-specific conventions

   The prompt does not explicitly ask the LLM to avoid redundancy in its comments. It's possible that a better constructed prompt would include guidance on how to avoid redundant comments, especially as the this prompt asks the LLM to consider the needs of newer teammates.

2. Which of the following is NOT a purpose of code documentation.

   - To help others learn to use your code
   - To prevent technical debt
   - To increase overall code quality
   - [✓ To decrease the length of a piece of code]

   Developing documentation may not lead to code that is shorter and, if anything, may increase the length of code. Hopefully, however, well-written comments and documentation will improve the intelligibility of a piece of software.

3. What should inline comments focus on explaining?

   - Redundant explanations of code functionality.
   - Every single line of code.
   - Basic syntax rules of the programming language.
   - [✓ Complex or non-obvious parts of the code.]

   Inline comments should clarify complex or non-obvious parts of the code to enhance understanding without overwhelming readability.

4. Which of the following is a best practice for writing effective docstrings?

   - Use stylistic variation to distinguish human and LLM-generated docstrings.
   - Write docstrings, potentially with the support of an LLM, in a clear and concise format
   - Exclude descriptions of parameters to avoid lengthy docstrings
   - Use an LLM to help vary the docstring format used between different methods to ensure the compatibility with the greatest number of external documentation tools

   Effective docstrings should be clear and concise to make the code easier to understand.

5. Which of the following is a best practice of writing code documentation?

   - Write documentation with your own understanding of the system in mind
   - Document code line by line to ensure no concepts are missed
   - Align documentation with language-specific conventions to ensure consistency within your project and compatibility with documentation tools.
   - Repeatedly document the same concept to ensure readers have multiple opportunities to understand the system

   Most languages have documentation conventions that help readers of your code know what to expect and may help documentation tools process your comments to build external documentation.