Coding Standards
Naming Conventions:
- Class names should follow the CapsWords convention (ex. 'Hangman')
- Use names that accurately explain the purpose of both functions and variables
- Avoid excessively long names (>24 characters)
- Variable names should be lowercase, with underscores separating words (ex. word_to_guess)
- Constants should be in uppercase with underscores separating words (ex. 'WORDS')

Style and Spacing:
- Use a two-space tab standard indentation
- Use spacing between functions and lines of code when necessary to improve readability, but avoid excess spacing
- Use comments to explain the purpose of functions and other lines of complex code when necessary
  - Inline comments:
    - def get_word(): # function to get random word
  - Other comments:
    - def get_word():
      # function to get random word

Function Standards:
- Functions should have clear and singular purposes
- Avoid overly complex logic within the functions – make helper functions if needed to make it easier for others to follow along
- Ensure consistent error handling, such as input validation

Use of Global Variables:
- Minimize the use of global variables – only use them if/when necessary

Making Pull Requests:
- Include clear and concise descriptions of the changes made in the pull request
- Reference relevant issues or give context for the changes made
- Ensure that all tests are passed, and no new issues are introduced

Reviewing Pull Requests:
- Provide constructive feedback with an emphasis on clarity, readability, and adherence to coding standards
- Confirm that the changes made pass all tests and no new bugs are introduced
- Check for potential improvements or optimizations that could be made
- Ensure the code aligns with the project's overall structure and goals

Testing Requirements:
- Include comprehensive test coverage for new features and modifications
- Ensure tests cover all scenarios, such as valid inputs and edge cases

- Document testing procedures and requirements for other contributors
- Run tests regularly to catch errors early on