

1.Naming Conventions

- Class name with capital word (Book).
- Function and variable name should be in small case if requires add underscore (borrow_book).
- All capital letters with underscores separating words(NUMBER_OF_DAYS and TOTAL).

2.DRY Principle(Don't repeat Yourself)

- Avoid code duplication at all cost
- Proposed solution: decorators

3.Customized exception

- Makes the code more readable
- Easier to debug
- More maintainable

4. With statement

- "With" statement, is better syntax to open the files
- Exceptions handling automatically
- Closing the file soon after task done automatically
- No need to write f.close() explicitly.

5.Don't keep Commented code

- It's looks ugly
- Bad habit of writing code
- Sonar will identify these kind of issues

6.Things to avoid

- Avoid passing Boolean into a function
- Returning null values
- Don't use a comment when you can use a well named Function or a Variable

7. Replace Magic Number with Symbolic Constant

- It makes harder to understand the program
- Refactoring the code also confusing
- Yet more difficulties arise when you need to change magic number.

8.Single responsibility principle

- One function must serve one purpose
- One class serve one purpose
- If not refactor the code

9.Avoid the cyclomatic complexity

- High cyclomatic complexity increases the risk of the presence of defects in the code due to increased difficulty in its testability, readability, and maintainability
- Below 4 is considered good;
- 5 and 7 is considered medium;
- 8 and 10 is high;
- above that is extreme complexity.

10. "small is better"

- Classes should also be "smaller than small",200 lines is a good limit for a class
- 10 classes per package
- Class should consist of average of 12 methods
- Methods should not have more than an average of 30 code lines (not counting line spaces and comments).

11.Don't be innovative; be simple.

- Be consistent. Use one word for similar functions. Don't use "get" in one class and "fetch" in another
- Don't hesitate to use technical terms in names. Go ahead, use the technical term. Your fellow programmer will understand it. For example, "jobQueue" is better than "jobs".
- Use consistent naming conventions. Always use uppercase, and separate words with underscores.

12. Keep arguments to fewer than or equal to three

- If arguments are greater than three, then you must think to refactor the function into a class.
- Indentation is also important. Code must use four spaces for indents, not the tab key.
- Class should consist of average of 12 methods
- Methods should not have more than an average of 30 code lines (not counting line spaces and comments).

13. Review your code yourself.

- Review your code once in a while. You might something new every time.
- Try to review with others as well before pushing the code to master.
- Don't hesitate to consider suggestions and improve it.
- Write unit tests to make sure existing code is not breaking.