Clean Code Development (CCD) Cheat Sheet

1. Meaningful Names

- Use descriptive, unambiguous names for variables, methods, and classes.
- Avoid single-letter variable names unless used in a clear context (e.g., loop counters).
- Make names reflect their purpose.

2. Single Responsibility Principle

- Ensure each class or method has only one reason to change.
- Break down large classes into smaller, cohesive ones with distinct responsibilities.

3. Avoid Magic Numbers

- Replace literal numbers with named constants to improve readability and maintainability.

4. Keep Functions Small

- A function should do one thing and do it well.
- Limit functions to 20 lines or fewer.
- Break long or complex logic into helper functions.

5. Use Comments Sparingly

- Comment why a decision was made, not what the code does.
- Clean and self-explanatory code should reduce the need for comments.

6. Follow the DRY Principle

- Do not Repeat Yourself. Extract repeated code into reusable methods or classes.

7. Write Clean Tests

- Test code should be as clean as production code.

- Use descriptive names for test cases and assertions.

- Ensure your tests are easy to understand and maintain.

8. Limit Dependencies

- Minimize coupling between modules.

- Use dependency injection where applicable for better flexibility and testability.

9. Encapsulation and Information Hiding

- Use private fields and methods to limit exposure.

- Only provide a public API for what is necessary.

10. Graceful Error Handling

- Handle errors using try-catch blocks or validation methods.

- Avoid using exceptions for normal flow control.

- Provide meaningful error messages for debugging.

This cheat sheet is a compact guide to writing clean, maintainable, and efficient code.