Clean Code Development (CCD) Cheat Sheet

1. Meaningful Names

- Use clear, descriptive names for variables, methods, and classes.
- Avoid single-letter names unless it’s obvious (like for loop counters).

2. Single Responsibility Principle

- Each class or method should do one thing only.
- Break up large classes into smaller, focused ones.

3. Avoid Magic Numbers

- Replace numbers with named constants to make code clearer.

4. Keep Functions Small

- Functions should do one job and do it well.
- Keep functions under 20 lines.

5. Comment When Necessary

- Use comments to explain "why" something is done, not "what" is done.
- Code should be clear enough to explain itself.

6. Follow DRY Principle

- "Don’t Repeat Yourself."
- Reuse code by putting common actions in methods or classes.

7. Write Clean Tests

- Test code should be as clean as the main code.
- Name test cases and checks clearly.

8. Limit Dependencies

- Keep classes and modules loosely connected.
- Use dependency injection when needed to make code more flexible.

9. Encapsulation and Information Hiding

- Keep details hidden; show only what’s necessary.
- Use private methods and fields to protect data.

10. Error Handling

- Handle errors clearly with checks and try-catch blocks.
- Don’t use exceptions for regular program flow.

11. Consistent Formatting

- Use the same indentations, spaces, and line breaks throughout the code.
- Keep the style uniform to make code easier to read.

12. Avoid Global State

- Don’t use global variables or shared state.
- Make state specific to classes or methods.

13. Use Abstractions

- Hide complexity with simpler interfaces or classes.
- Use abstractions to make code easier to maintain.

14. Refactor Regularly

- Regularly update and improve the code.
- Refactor when features change or new ones are added.

15. Test Coverage

- Ensure most code is tested to catch issues early.
- Use unit tests for small functions and integration tests for combined features.