Top 13 Clean Code Principles (based on Uncle Bob's book):

1. Meaningful Names

Use clear, descriptive names for variables, functions, and classes.

Example: age is better than a.

2. Small Functions

Functions should be short and focused.

Ideally 5–15 lines and doing one thing only.

3. Do One Thing

A function or class should have one responsibility only (Single Responsibility Principle).

4. Descriptive Function Names

Functions should be named after what they do.

Example: calculateInvoiceTotal() is better than calc().

5. Avoid Unnecessary Comments

Write code that explains itself.

Use comments only when necessary for clarification.

6. DRY (Don't Repeat Yourself)

Avoid duplicating logic. Reuse code through functions or modules.

7. Proper Formatting & Indentation

Well-indented and neatly formatted code is easier to read and maintain.

8. Clear Error Handling

Use try/except in a clean, structured way. Don't let error handling hide your main logic.

9. Avoid Magic Numbers & Strings

Use constants with meaningful names instead of hard-coded values.

Example: MAX_RETRIES = 3 instead of writing if attempts > 3.

10. Separate Concerns

Divide your code into layers or logical sections (e.g., UI, logic, data).

11. Keep It Simple (KISS)

Avoid over-complicating things. The simplest solution that works is usually the best.

12. Write Unit Tests

Clean code is testable. Writing tests ensures reliability and future-proofing.

13. Refactor Often

Regularly improve and clean up your code, even if it already works.