Clean Code principles:

1. Single Responsibility Principle (SRP): A class or function should have only one reason to change, i.e., it should have only one responsibility.
2. Open/Closed Principle (OCP): Software entities (classes, modules, functions) should be open for extension but closed for modification. This means you can add new functionality without modifying existing code.
3. Liskov Substitution Principle (LSP): Objects of a superclass should be replaceable with objects of a subclass without affecting the correctness of the program.
4. Interface Segregation Principle (ISP): Clients should not be forced to depend on interfaces they do not use. This principle encourages creating specific interfaces for specific client needs.
5. Dependency Inversion Principle (DIP): High-level modules should not depend on low-level modules; both should depend on abstractions. Abstractions should not depend on details; details should depend on abstractions.
6. Don't Repeat Yourself (DRY): Avoid duplication in code. Instead, extract common functionality into reusable components.
7. Keep It Simple and Small (KISS): Strive for simplicity in code. Keep functions and classes small and focused on a single task.
8. Separation of Concerns (SoC): Split your code into distinct sections, each addressing a separate concern.
9. Command-Query Separation (CQS): Methods should either change state (commands) or return information (queries), but not both.
10. Single Level of Abstraction (SLA): Each function or method should operate at a single level of abstraction, meaning all the statements should be at the same level of detail.
11. Consistency: Use consistent naming, formatting, and coding conventions throughout your codebase.
12. Avoid Premature Optimization: Focus on writing clean and clear code first. Optimize for performance only when necessary and based on actual profiling data.
13. Code Smells: Be aware of "code smells," which are indicators of potential problems in code quality. Some common code smells include long functions, large classes, and excessive commenting.