**Software Development Principles**

**SOLID Principles**

1. SRP (Single Responsibility Principle)

A class/function/module should have only one responsibility.

1. OCP (Open/Closed Principle)

Open for Extension, but Closed for Modification (Encapsulation, Inheritance, and composition)

1. LSP (Liskov Substitution Principle)

Subclass should behave in the same way as the objects of your superclass.

1. ISP (Interface Segregation Principle)

Make your interfaces small and focused.

1. DIP (Dependency Inversion Principle)

The High-level modules should depend on low-level modules only by interfaces. Decoupling.

**KISS (Keep It Simple, Stupid)**

Avoiding unnecessary complexity will make your system more robust, easier to understand, easier to reason about, and easier to extend.

**DRY (Don’t Repeat Yourself)**

Inheritance and Composition. Database Normalization.

**YAGNI (You Aren’t Gonna Need it)**

Always implement things when you actually need them. “Premature optimization is the root of all evil”.