SOLID: The 5 Principles of Object Oriented Design

S - **S**ingle-responsibility Principle

O - **O**pen-closed Principle

L - **L**iskov Substitution Principle

I - **I**nterface Segregation Principle

D - **D**ependency Inversion Principle

1. Single-responsibility Principle (SRP) states:

A class should have one and only one reason to change, meaning that a class should have only one job.

2. Open-closed Principle (OCP) states:

Objects or entities should be open for extension but closed for modification.

3. Liskov Substitution Principle states:

subclasses should be substitutable for their base classes.

This means that, given that class B is a subclass of class A, we should be able to pass an object of class B to any method that expects an object of class A and the method should not give any weird output in that case.

4. Interface segregation principle states:

A client should never be forced to implement an interface that it doesn't use, or clients shouldn't be forced to depend on methods they do not use.

5. Dependency inversion principle states:
   Entities must depend on abstractions, not on concretions. It states that the high-level module must not depend on the low-level module, but they should depend on abstractions.