Single Responsibility Principle (SRP)

SRP is that every class, module, or function in a program should have one responsibility in a program. As a commonly used definition, "every class should have only one reason to change".

Open–Closed Principle (OCP)

The open-closed principle states that software entities should be open for extension, but closed for modification.

Liskov Substitution Principle (LSP)

The Liskov substitution principle simply implies that when an instance of a class is passed/extended to another class, the inheriting class should have a use case for all the properties and behavior of the inherited class.

Interface Segregation Principle (ISP)

The interface segregation principle states that the interface of a program should be split in a way that the user/client would only have access to the necessary methods related to their needs.

Dependency Inversion Principle (DIP)

High-level modules should not import anything from low-level modules. Both should depend on abstractions (e.g., interfaces). And, Abstractions should not depend on details. Details (concrete implementations) should depend on abstractions.

SOLID Definition – the SOLID Principles of Object-Oriented Design Explained