SOLID principles are design principles that enable us to manage most of the software design problems.

S - Single Responsibility (Every class or similar structure, in your code should have only one job to do)

O - Open Closed (A software module/class is open for extension and closed for modification)

L - Liskov Substitution (Should be able to use any derived class instead of a parent class and have it behave in the same manner without modification)

I – Interface Segragation(that clients should not be forced to implement interfaces they don't use. Instead of one fat interface, many small interfaces are preferred based on groups of methods, each one serving one submodule.)

D -Dependency Inversion (The Dependency Inversion Principle (DIP) states that high-level modules/classes should not depend on low-level modules/classes. Both should depend upon abstractions. Secondly, abstractions should not depend upon details. Details should depend upon abstractions.)