SOLID PRINCIPLES

In software engineering, **SOLID** is a mnemonic acronym for five design principles intended to make object-oriented designs more understandable, flexible, and maintainable.

The principles: are a subset of many principles promoted by American software engineer and instructor Robert C. Martin first introduced in his 2000 paper Design Principles and Design Patterns discussing software rot.

The **SOLID** acronym was introduced later, around 2004, by Michael Feathers.

Although the **SOLID** principles apply to any object-oriented design, they can also form a core philosophy for methodologies such as agile development or adaptive software development.

The **SOLID** ideas are:

1. **The Single-responsibility principle:**

   "There should never be more than one reason for a class to change." In other words, every class should have only one responsibility.

2. **The Open–closed principle:**

   "Software entities ... should be open for extension, but closed for modification."
   Modification means changing the code of an existing class, and extension means adding new functionality.

3. **The Liskov substitution principle:**

   "Functions that use pointers or references to base classes must be able to use objects of derived classes without knowing it."

   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. **The Interface segregation principle:**

"Clients should not be forced to depend upon interfaces that they do not use."

The principle states that many client-specific interfaces are better than one general-purpose interface. Clients should not be forced to implement a function they do no need.

5. **The Dependency inversion principle:**

The Dependency Inversion principle states that our classes should depend upon interfaces or abstract classes instead of concrete classes and functions.

"Depend upon abstractions, [not] concretions."