SOLID (object-oriented design)

**S(SRP): Single responsibility principle**

A class should have only a single responsibility (i.e. only one potential change in the software's specification should be able to affect the specification of the class)

**O(OCP): Open/closed principle**

“software entities … should be open for extension, but closed for modification.” ..

**L(LSP): Liskov substitution principle**

“objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program.” See also design by contract.

**I(ISP): Interface segregation principle**

“many client-specific interfaces are better than one general-purpose interface.”[8]

**D(DIP): Dependency inversion principle**

one should “depend upon abstractions, [not] concretions.”[8]

**Notes:** Single Responsibility (SRP), Open/Closed (OCP), Liskov's Substitution,

Interface Segregation, and Dependency Inversion.

**Five agile principles that should guide you every time you need to write code.**

**SRP:**   
“A Class should have only one reason to change”

**OCP:**   
In object-oriented programming, the open/closed principle states "software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification" that is, such an entity can allow its behaviour to be extended without modifying its source code.

**LSP:**   
If S is a subtype of T, then objects of type T may be replaced with objects of type S (in other words, objects of type S may substitute objects of type T) without altering any of the desirable properties of that program (correctness, task performed, and so on).

**s=subtype of T**

***ISP:***The **interface-segregation principle** (**ISP**) states that no client should be forced to depend on methods it does not use.

**DIP:**  
 ***A. High-level modules should not depend on low-level modules. Both should depend on*** [***abstractions***](https://en.wikipedia.org/wiki/Abstraction_(computer_science))***.***

***B. Abstractions should not depend on details. Details should depend on abstractions.***

**High-level module, low-level module, details depend on Abstraction.**