**What is Solid?**

SOLID are five basic principles which help to create good software architecture. SOLID is an acronym where: -

* S stands for SRP (Single responsibility principle
* O stands for OCP (Open closed principle)
* L stands for LSP (Liskov substitution principle)
* I stand for ISP (Interface segregation principle)
* D stands for DIP (Dependency inversion principle)

**SRP (Single Responsibility Principle)**

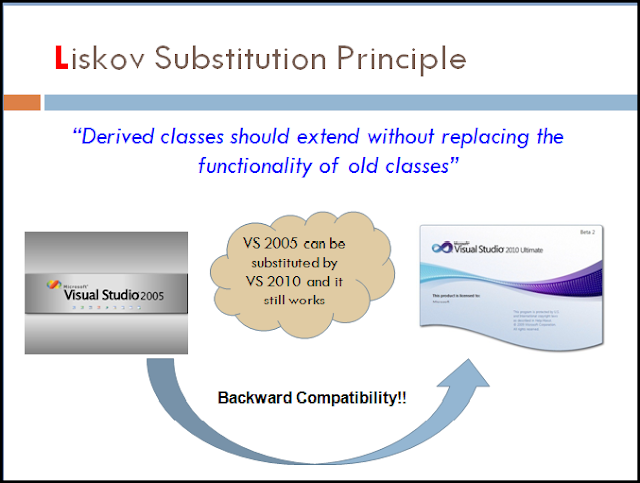
Single Responsibility principle says that a class should have only one responsibility, rather dealing multiple activities.

**Open Closed Principle**

Stop modifying existing class, rather create new class and implement our logic.

**Liskov Substitutional Principle:**

LISKOV principle says that parent should easily replace the child object. So, to implement Liskov we should extend derived classes with our new functionalities.



**Interface Segregation Principle:**

Interface segregation principle says that, rather modifying existing interface, we should create new one to avoid unnecessary object utilization in our classes.

**Dependency inversion Principle:**

High level modules should not depend on low level modules but should depend on abstraction.

