SOLID Introduction

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

The term SOLID is an acronym for five design principles intended to make software designs more understandable, flexible and maintainable

Acronym

S-Single Responsibility Principle (SRP)

O- Open close Principle (OSP)

L- Liskov substitution Principle (LSP)

I – Interface Segregation Principle (ISP)

D- Dependancy inversion Principle (DIP)

SINGLT RESPOSIBILITY PRINCIPLE – Every module or a class should have a single responsibility to implement.i,e a class should have only one reason to change

OPEN CLOSE PRINCIPLE – The code should be open for extension but closed for modification.

LISKOV SUBSTITUTION PRINCIPLE – If a program is using a base class the the reference to the base class should be replace by the reference of that of object class without affecting the functionality. Ie base class objects should replace parent class objects.

Interface segregation – A client should never be forced to implement a interface which fe/she does not use. Instead we can create various interfaces for various tasks(breaking down ).

DEPENDANCY INVERSION PRINCIPLE – High level modules should not depend on the low level modules but rather both should depend on abstraction.