**SOLID Design Principles in Java**

**S – Single Responsibility Principle.**

**O – Open Closed Principle.**

**L – Liskov Substitution Principle.**

**I – Interface Segregation Principle**

**D – Dependency Inversion Principle.**

**S – Single Responsibility Principle.**

Every Class should Have Single Responsibility.

For Example

We have Class called Bank Service and in this we have so many methods like

Deposit, WithDraw, PrintPassbook, getLoanIntrestInfo, SendOTP

**O – Open Closed Principle.**

This Principle says that software entity like classes, modules, functions should be open for extension but close for modification.

**L – Liskov Substitution Principle.**

In this Deriver or childe classes must be substitutable for their base or parent classes.

In other words class A is a subtype of class B then we should be able to replace B with A without interrupting the behavior of program.

**I – Interface Segregation Principle**

This principle state that do not force any client to implement an interface which is irrelevant to them.

For example if Interface having 5 method and Class A do not want to override all the method than we cant force to override the method.

**D – Dependency Inversion Principle.**

This principle state that we must use abstraction (abstract classes and interfaces) instead of concrete implementation. High level module should not depend on the low-level module but both should depend on the abstraction.