**SOLID Principles:**

These Principles are used in OOD(Object Oriented Design) which helps us to keep the code

* modular
* maintainable
* less complex
* easy to understand
* avoid duplication

**Acronym of SOLID**

S - Single Responsibility Principle

O - Open/Closed Principle

L - Liskov Substitution Principle

I - Interface Segregation Principle

D – Dependency Inversion Principle

**Single Responsibility Principle:**

* This principle states that **each class should have** **one responsibility, one single purpose**. This means that a class will do only one job, which leads us to conclude it should have **only one reason to change**
* Any Method/Function/Class which we create should have one responsibility and should only have one reason to change

Advantages:

* Adding multiple functionalities to single class can lead to complexity and hard to maintain
* Changing one functionality can effect other functionalities because they exist in a single class

**Examples:**

**Example 1:** We need to develop a Feature where we can manipulate a given text in different ways and Print the text in different ways

**Solutions:**  This feature can be developed in many ways, mentioning some below:

1. Developing both the features in a single place
2. Break Down the Feature into multiple parts where each part does one job

* One Class does the job of Text Manipulating
* One Cass does the job of Print the text

**Developing both the features in a single place**