OOP: Helps us to provide a way to write and polish our code in a better way.

OOD: We need to think beyond this so that we can have better design SOLID helps us with this.

S - Single Responsibility principle

O - Open and close principle

L - Liskov sunstitution principle

I - Interface segregation principle

D – Dependency inversion principle

SRP single responsibility principle:

As name suggest class should have a single responsibility.

There should be only one reason to changes the class. Assigning more behaviour to the class lead to changing the class very now and then.

Eg: I have a person table in which I have to perform CRUD operations this has to be designed separately so that not all operations are done as apart of one class.

Single focused funtionality

Eg: Journals with check list

Maintaining the journals

Maintaining the persistence of the Journals.

Open close principle:

Eg:1. Product with name, size ,color

ISpecification, Ifilter

Filter product based on size and Then color, next is based on both

2. Invoice validation

Add a validator class and perform validation for each