Java Design Principles

SOLID:

S - single responsibility principle(SRP)

A class should have only one reason to change meaning that it should have only one responsibility

 O - open closed principle (OCP)

open for extension but closed to edits and modification.

L - Liskov substitution principles(LSP)

 subclasses of a superclass should be able to replace superclass objects without affecting the program's correctness.

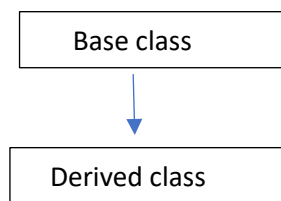**I** - interface segregation principle(ISP)

 suggests that interfaces should be tailored to the specific needs of the classes that implement them.

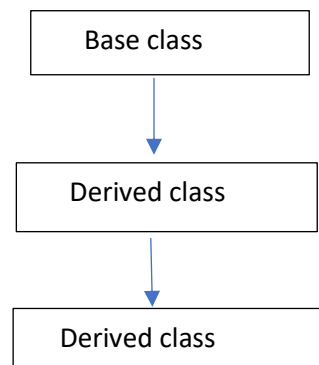**D** - Dependency Inversion principle(DIP)

DIP tells us that every dependency in the design should target an interface or an abstract class.

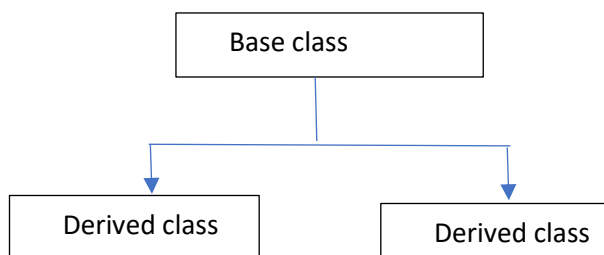# Oops(object oriented programming language) concept in java

- o **Inheritance**: Deriving one class from another class is called inheritance
  - Single inheritance
  - Multi-level inheritance
  - Hierarchical
- o Single inheritance:

```
        ┌─────────────────┐
        │   Base class    │
        └─────────────────┘
                 │
                 ▼
        ┌─────────────────┐
        │  Derived class  │
        └─────────────────┘
```

- o Multi level inheritance:

```
        ┌─────────────────┐
        │   Base class    │
        └─────────────────┘
                 │
                 ▼
        ┌─────────────────┐
        │  Derived class  │
        └─────────────────┘
                 │
                 ▼
        ┌─────────────────┐
        │  Derived class  │
        └─────────────────┘
```

- o Hierarchical inheritance:

```
              ┌─────────────────┐
              │   Base class    │
              └─────────────────┘
                       │
          ┌────────────┴────────────┐
          ▼                         ▼
  ┌─────────────────┐       ┌─────────────────┐
  │  Derived class  │       │  Derived class  │
  └─────────────────┘       └─────────────────┘
```

- o Multilevel inheritance
- o Hybrid inheritance

```
        ⎫─────────⎰─────────⎰
                  │
              Interface
```

Polymorphism:

Polymorphism in Java is the task that performs a single action in different ways.

Eg: A person who at the same time can have different characteristics. A man at the same time is a father, a husband, and an employee.

Abstraction:

Data abstraction is the process of hiding certain details and showing only essential information to the user

Eg: Making coffee with a coffee machine

Assosciation:

It is a relationship between class and object

1.aggregation- Is a relationship

  1.one to one

  2.one to many

  3.many to one

  4.many to many

2.composition- has a relationship

Encapsulation: binding data and methods together in a single unit