**Java Programming 🡪 By Abdul Bari**

* **Principle of OOPS**

1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism
5. Abstraction: showing only required details and hiding the internal details of anything.

Data **abstraction** is the process of hiding certain details and showing only essential information to the user.  
Abstraction can be achieved with either **abstract classes** or [**interfaces**](https://www.w3schools.com/java/java_interface.asp)

Link of the website: https://www.w3schools.com/java/java\_abstract.asp#:~:text=The%20abstract%20keyword%20is%20a,does%20not%20have%20a%20body.

1. Encapsulation: We can store bundle of thing into the box just like that encapsulate the thing into the one place.

**Encapsulation** is defined as the wrapping up of data under a single unit. It is the mechanism that binds together code and the data it manipulates. Another way to think about encapsulation is, that it is a protective shield that prevents the data from being accessed by the code outside this shield.

Link of the website: https://www.geeksforgeeks.org/encapsulation-in-java/

1. Inheritance: something to resuse from existing one like we modify the mobile function with the new update by mobile company.

Inheritance is an important pillar of OOP(Object-Oriented Programming). It is the mechanism in java by which one class is allowed to inherit the features(fields and methods) of another class.

Link of the website: <https://www.geeksforgeeks.org/inheritance-in-java/>

1. Polymorphism: you need to write or overwrite the existing code then we will use polymorphism approach

Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance.

Link of the Website: https://www.w3schools.com/java/java\_polymorphism.asp#:~:text=Polymorphism%20means%20%22many%20forms%22%2C,methods%20to%20perform%20different%20tasks.

1. Generalization: Like we watching TV But we do not rely on which brand of TV is thier. This is similar to Polymorphism.

Converting a subclass type into a superclass type is called ‘**Generalization**‘ because we are making the subclass to become more general and its scope is widening. This is also called **widening or up casting**. Widening is safe because the classes will become more general.

Link to the website: https://www.geeksforgeeks.org/generalization-and-specialization-in-java/#:~:text=Converting%20a%20subclass%20type%20into,classes%20will%20become%20more%20general.

1. Specialization: Like we create a new thing from old thing that’s using a existing properties.

Converting a super class type into a sub class type is called ‘**Specialization**‘. Here, we are coming down from more general form to a specific form and hence the scope is narrowed. Hence, this is called **narrowing** or **down-casting**.

* Class vs Object:

Everything in the world we define properties and behaviour

1. Properties
2. Behaviour