



# Java Language

## Notes

# DEMO

For more PDFs and  
notes visit my  
telegram page..

Named "**BeingPro33**"

Handwritten Notes



@beingpro33



## Object Oriented Programming System (OOPS) -



Object oriented programming is a programming paradigm that revolves around the concept of objects, which can contain data and methods to manipulate the data.

### \* Features of OOPS -

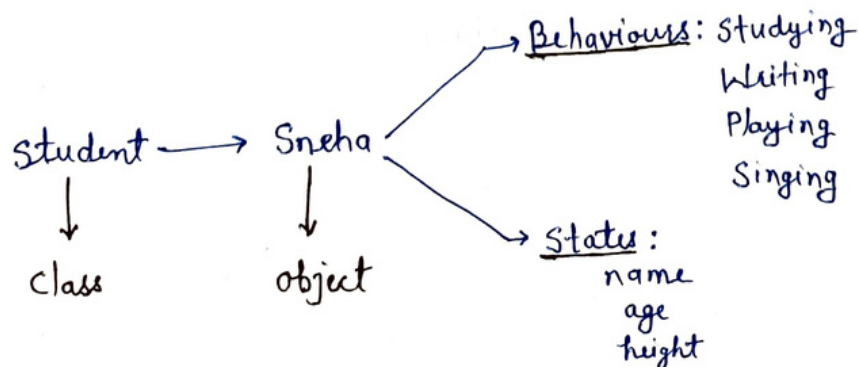
1. Inheritance
2. Abstraction
3. Encapsulation
4. Polymorphism

### \* Class -

A class is a blueprint or a template for creating objects that defines a set of variables, methods and properties that are common to all objects of that class.

### \* Object -

An object is an instance of a class or an entity which gets created using class and it represents the state and behaviour.



# Being Pro

## \* Constructor -

A constructor is a special type of member function that is called automatically when an object is created.

- Constructor is required for initialization of properties at the time of construction of an object.
- Every class will have a default constructor provided by java compiler.
- Constructor will not have any return type.

Syntax :

```
public constructorName (arg/Noarg)
{
    // Body of constructor
}
```

## \* Rules for defining Constructor -

- Constructor can be public, private, protected or default.
- Constructor can not be static, non-static, final or abstract.
- Constructor name must be same as that of class name.
- It does not have any return type not even void.
- Constructor can be with arguments or without arguments.

## Inheritance

- \* It is process of acquiring features of an existing class into a new class.
- \* The class that inherits properties is called the subclass or derived class or child class.
- \* And the class that provides properties is called the superclass or base class or parent class.
- \* In java, "extends" keyword is used to establish an inheritance relationship b/w two classes.

### Example -

→ A cylinder can acquire all the properties of circle plus it can have extra features, where we can write a class cylinder inheriting from class circle.

```
class Circle // Parent class
{
    public double radius;
    public double area()
    {
        return Math.PI * radius * radius;
    }
    public double perimeter()
    {
        return 2 * Math.PI * radius;
    }
    public double circumference()
    {
        return perimeter();
    }
}
```

## Abstraction

\* It is the process of hiding the internal implementation and showing the necessary data to the user, is called abstraction.

Eg:- Sending messages, we just type the text and press on the send button. We don't know the internal processing, how it is being send.

\* In java, abstraction can be achieved in two ways-

- i) Using abstract class
- ii) Using interfaces

\* Abstract class -

If 'abstract' keyword is used before the class then it is called as abstract class.

→ If nothing is written before the class then it is called concrete class (Normal class that we write)

X → An abstract class will always have atleast one abstract method. X

\* Abstract method-

A method which is not having a body is known as Abstract method. and the method must be declared as abstract.

\* An abstract class can have abstract and non-abstract method.

## Interface

- \* An interface is a collection of abstract methods and constants, but without any implementation
- \* It is a way to achieve abstraction, as it allows the programmer to focus on the behaviour of an object rather than its implementation.
- \* An ~~inf~~ interface has to be represented with 'interface' keyword.

Syntax:

```
interface interfaceName
{
    // Body of interface
}
```

- \* All the methods of interface are by default public and abstract whether we write or don't write.
- \* In interface, we can not create an object of interface because all methods are by default abstract.
- \* But we can create a reference of <sup>interface</sup> variable and can be assigned the object of that class which is implemented.
- \* A class can extend from only one class at a time
- \* But a class can implement multiple interface at a time.

## Polymorphism

- \* Polymorphism is a concept in which we can execute a single operation in different ways.
- \* Polymorphism is that which is used to reduce the no. of functions to be remember.
- \* The word 'Polymorphism' is derived from two greek words: 'Poly' and 'morphi'.
- \* The word 'Poly' means many and 'Morphi' means forms. So polymorphism means 'many forms'.
- \* There are two types of polymorphism in java -
  - i) Compile time Polymorphism (Overloading)
  - ii) Run time polymorphism (Overriding)
- i) Compile time polymorphism:
  - It is also known as static polymorphism or early binding.
  - Compile time polymorphism can be achieved by overloading.

Eg: Method overloading, ~~Cost~~ Constructor overloading