

### Topics to be Covered

\* OOPS (Object Oriented Programming)

→ Functions / Methods

Inheritance

↳ Method Overriding

### Functions

In Java, functions are called methods when they are defined within a class. Method encapsulate behaviour and allow objects of a class to perform specific actions or computations.

- Method Declaration : Methods are declared within a class and have a signature that specifies the access modifier, return type, method name and parameters (if any) . . .

For Ex -

```
public class calculator {  
    public int add(int a, int b) {  
        return a + b;  
    }  
}
```

- Method Invocation : Once a method is defined, it can be invoked on objects of the class using the dot operator.

- Method Overloading : Java supports method overloading, means that multiple methods can have the same name but different parameters.

↳ The compiler determines the correct method to invoke based on argument provided.

- Static Methods : Java allows the declaration of static Methods that belong to the class itself, rather than specific Objects.

↳ Static Methods can be invoked directly on the class without creating an instance.

## Inheritance :

Inheritance is a mechanism in Java that allows one class to inherit properties and behaviours from another class.

### - SuperClass and Subclass :

```
public class Animal {
```

```
    public void eat() {
```

```
        System.out.println("We are Strong");
```

```
    }
```

```
    }
```

```
public class Dog extends Animal {
```

```
    public void bark() {
```

```
        System.out.println("Dog is barking");
```

```
    }
```

## Types of Inheritance

1. Single Inheritance: is the simplest form of inheritance where a subclass extends only one superclass.

2. Multilevel Inheritance: Multilevel inheritance occurs when a subclass extends another subclass, creating a chain of inheritance.

- Each subclass inherits the methods of its superclass, and new methods can be added to each subclass.

### 3. Hierarchical Inheritance :

This inheritance involves multiple subclasses extending a single superclass.

### 4. Hybrid Inheritance :

Hybrid Inheritance combines multiple types of inheritance.

- It involves a mix of single inheritance, multilevel inheritance and multiple inheritance through 'interfaces'.

## Interfaces

In Java, an interface is a reference type that defines a contract for classes to follow.

- It provides a way to achieve abstraction and establish a common set of methods that implementing class must adhere to.

### - Declaration

An Interface is declared using the 'interface' keyword.

- Defines a collection of Abstract methods (methods without a body) and constant variables (implicitly 'public', 'static', 'final')

For Ex:-

```
interface Printable {
    void print();
}
```

g



In this example, the 'Printable' interface declares a single abstract method 'print()'. ↴

### - Implementation:

To use an interface, a class must implement it using the "implements" keyword.

↳ Implementing an interface means providing an implementation for all its declared methods.

```
[ class Printer implements Printable {
```

```
    public void print() {
```

```
        System.out.println("Printing....");
```

g

- Multiple Interface Implementation: A class can implement multiple interfaces, separate by comma (,).

```
class Document implements Printable, Readable {
```

g

- Interface Inheritance: An interface can extend one or more interfaces using the 'extends' keyword.

↳ This allows for the creation of hierarchy of interfaces.

[interface Shape {  
    void draw();  
}

interface Circle extends Shape {  
    void calculateArea();  
}

### Extras (Java 8)

- Default and Static Methods:

Starting from Java 8, interfaces can also contain default and static methods with method bodies.

interface Loggable {  
    void log();  
}

default void display() {  
    System.out.println("Displaying");  
}

static void info() {  
    System.out.println("Logging info");  
}