Day-15

OOPS Concepts – Inheritance

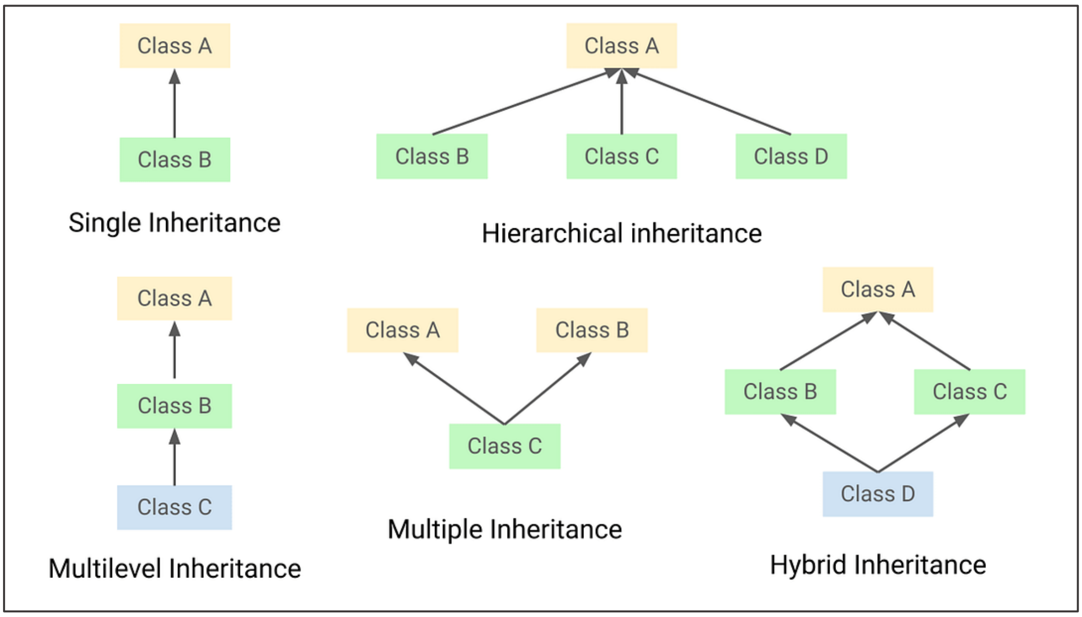
Inheritance

- **Inheritance** is a mechanism in Java where one class can acquire the properties (variables) and behaviours (methods) of another class.
- When a class inherits another class, it can use the variables and methods of the parent class without rewriting them.
- The class that is inherited is called the **parent class/super class/base class**, and the class that inherits is called the **child class/sub class/derived class**.
- We use **extends** keywords to inherit a class.

What is the Use of Inheritance?

- **Code Reusability**: Inheritance allows a class to reuse the variables and methods of another class, reducing duplication.
- **Method Overriding**: Subclasses can change the behaviour of methods inherited from the parent class.
- **Ease of Maintenance**: Changes made in the parent class reflect in the child class automatically, simplifying maintenance.

Types of Inheritance in Java



Single Inheritance:

A subclass inherits from a single parent class.

Syntax:

```
class Parent {
    // Parent class code
}

class Child extends Parent {
    // Child class code
}
```

Multilevel Inheritance:

A chain of inheritance where a class is derived from another class, which is also derived from another class.

Syntax:

```
class Grandparent {
    // Grandparent class code
}

class Parent extends Grandparent {
    // Parent class code
}

class Child extends Parent {
    // Child class code
}
```

Hierarchical Inheritance:

Multiple child classes inherit from the same parent class.

Syntax:

```
class Parent {
    // Parent class code
}

class Child1 extends Parent {
    // Child1 class code
}

class Child2 extends Parent {
    // Child2 class code
}
```

Multiple Inheritance:

- A class can inherit features from more than one parent class (**Not possible through Class in java**).
- It can be achieved through **interfaces**.
- A class can extend only one class.
- A class can implement multiple interfaces, with this we can achieve multiple inheritance.

Syntax (using interfaces):

```
interface Interface1 {
    // Interface1 methods
}

interface Interface2 {
    // Interface2 methods
}

class Child implements Interface1, Interface2 {
    // Child class implementing multiple interfaces
}
```

Hybrid Inheritance:

A combination of more than one type of inheritance.

Since Java does not support hybrid inheritance with classes directly, it is typically achieved using interfaces.

Why we cannot extend multiple classes in Java?

- **A class cannot extend more than one class at a time**. This is because of the possibility of **ambiguity**.
- For instance, if two parent classes have methods with the same signature, it would be unclear which method the child class should inherit, leading to conflicts.