UNIT-IV

Inheritance, Packages & Interfaces

Abstract

- Basics of Inheritance
- Types of inheritance

Basics of Inheritance

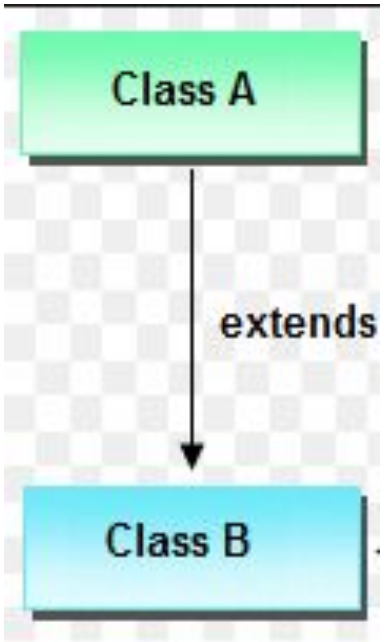
- Inheritance in Java is a mechanism in which, **one child class acquires all the properties and behaviors of a parent object.**

- It is an important part of OOPs (Object Oriented programming system).

- Inheritance represents the **IS-A relationship** which is also known as a ***parent-child* relationship.**

Basics of Inheritance

- **Advantage:**
  - **Method Overriding**
  - **Code Reusability.**
- **Importance Terms:**
  - **Super class / Base class / Parent class**
  - **Sub class / Derived class/ Child class**



Basics of Inheritance

- **Super Class:** The class whose features are inherited is known as super class(or a base class or a parent class).

- **Sub Class:** The class that inherits the properties of other class is known as sub class(or a derived class, extended class, or child class).

- The subclass can add its own fields and methods in addition to the super class fields and methods.

Basics of Inheritance

- **Syntax:**

```
class Subclass-name extends Superclass-name
{
  //methods and fields
}
```

- The **extends keyword** indicates that you are making a new class that derives from an existing class.

- It simply means “extends” the functionalities of a class.

Basics of Inheritance

```
class Calculation  // super class
{
    int z;
    public void addition(int x, int y)
    {   z = x + y;
        System.out.println(“ addition:"+z);
    }
}
```

Basics of Inheritance

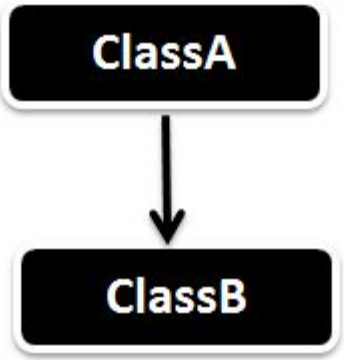
```
public class My_Calculation extends Calculation
{
    public void multiplication(int x, int y)
    {   z = x * y;
        System.out.println("Multiplication:"+z);
    }
    public static void main(String args[])
    {
        int a = 20, b = 10;
        My_Calculation demo = new My_Calculation();
    demo.addition(a, b);
        demo.multiplication(a, b);
    }
}
```

Types of Inheritance

- Single Inheritance
- Multiple Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance

Single Inheritance

- Single Inheritance is the simple inheritance of all, When a class extends another class(Only one class) then we call it as **Single inheritance.**
- **Class B** extends only one class **Class A.**
- Here **Class B** will be the **Sub class** and **Class A** will be one and **Super class**.



Multiple Inheritance

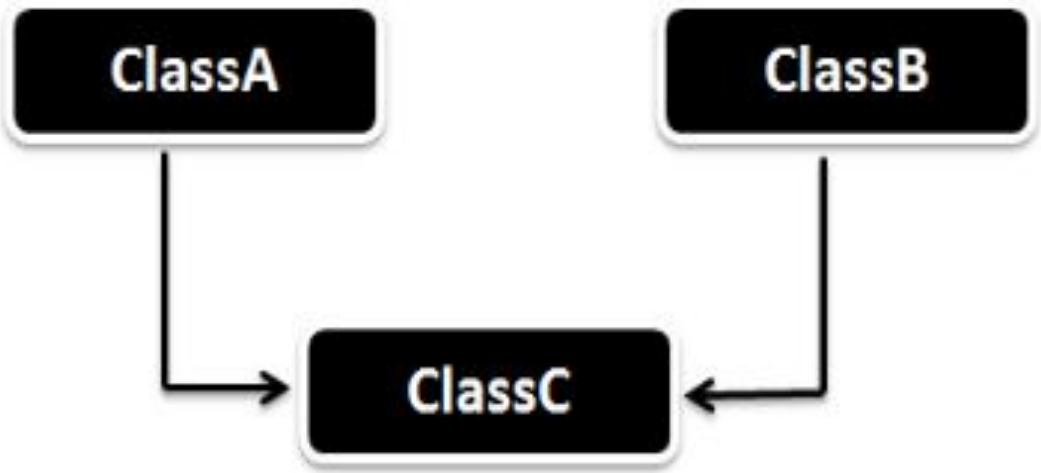
- **Multiple Inheritance** is nothing but **one** class **extending more than** one class.

- Multiple Inheritance is basically not directly supported by many **Object Oriented Programming** languages such as **Java, Small Talk, C# etc.. (C++ Supports Multiple Inheritance).**

- As the **Child** class has to manage the dependency of more than one **Parent** class.

Multiple Inheritance

- **But you can achieve multiple inheritance in Java using *Interfaces.***



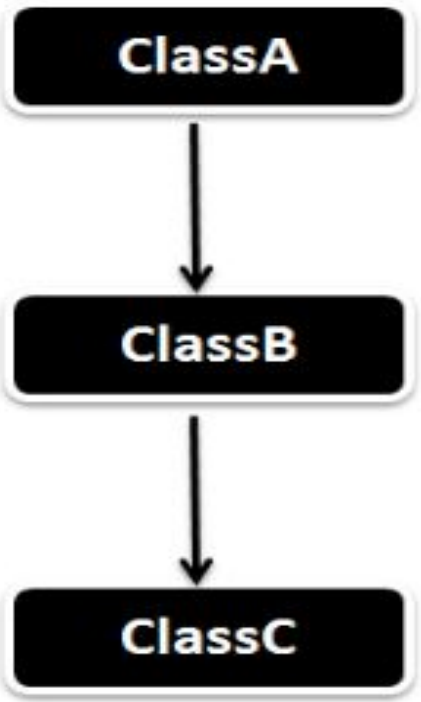
Multilevel Inheritance

- In **Multilevel Inheritance** a derived class will be **inheriting a parent class** and as well as the derived class **act as the parent class** to other class.

- **ClassB** inherits the property of **ClassA** and again **ClassB** act as a parent for **ClassC**.

- In Short **ClassA** parent for **ClassB** and **ClassB** parent for **ClassC**.

Multilevel Inheritance



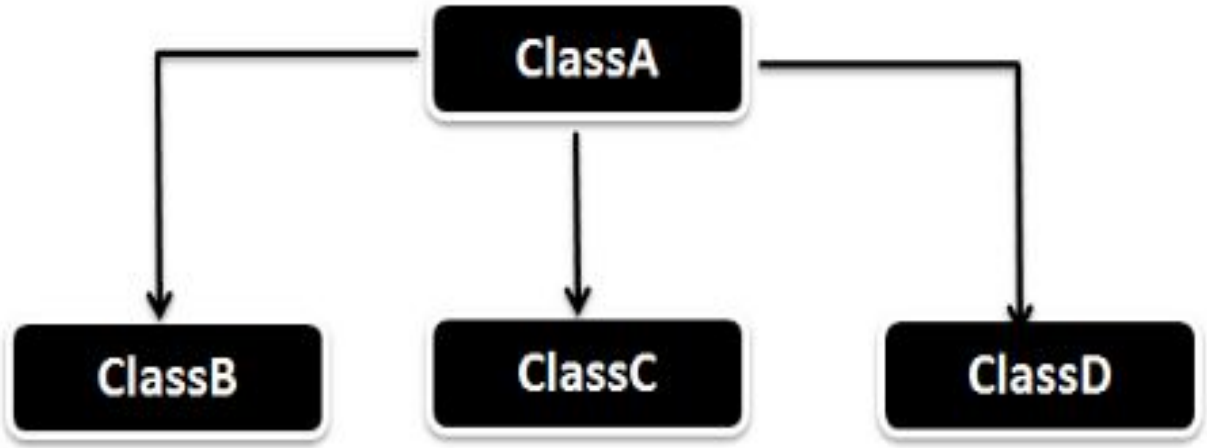
Hierarchical Inheritance

- In **Hierarchical inheritance** one parent class will be inherited by **many** sub classes.

- **ClassA** will be inherited by **ClassB, ClassC** and **ClassD**.

- **ClassA** will be acting as a parent class for **ClassB, ClassC** and **ClassD**.

Hierarchical Inheritance



Hybrid Inheritance

- **Hybrid Inheritance is the combination of both Single and Multiple Inheritance.**

- Hybrid inheritance is also not directly supported in Java only through interface we can achieve this.

- As you can **ClassA** will be acting as the **Parent** class for **ClassB & ClassC** and **ClassB & ClassC** will be acting as **Parent** for **ClassD.**

Hybrid Inheritance