

# **Basic Java**

## Non Primitive Data Type:-

- Array:-

An array is a non primitive datatype whose size is not fixed, An array is used to store multiple data of same data type.

Array always store homogenous data.

Ways to declare array :-

1. `int [] a = {5,4,6,7};`
2. `int a[] = {5,4,6,7};`
3. `int [] a;` - declaration (This type of array can have infinite value.)  
To assign value :- `a = new int [] {1,2,3,4};`

To declare an array with fixed size:-

- `int [] a = new int [20];` -(in this case we have declared an array with can only store 20 values of same data types.)

## Basic Concepts of OOps :-

- [Object](#)
- Class
- [Inheritance](#)
- [Polymorphism](#)
- [Abstraction](#)
- [Encapsulation](#)

### 1. Inheritance :-

Inheritance is an OOps concept where one class acquire the property of another class.

The class from which properties are acquired or inherited is called as Super Class/ Parent Class, and the class which inherits or takes the property is called as Sub Class/Child Class. We always use “extends” keyword for inheritance.

We can also say that inheritance is a parent child relationship.

For better understanding lets refer the diagram:-

## Without Inheritance

Class A {  
Method1,  
Method2,  
Method3}

Class B {  
Method 4}

Super Class  
/Parent Class

Class A {  
Method1,  
Method2,  
Method3}

Inheritance takes  
place between A &  
B

Class B {  
Method 1,  
Method 2,  
Method 3,  
Method 4}

Sub class /Child  
class

For ex:-

```
Class A {
```

```
    Public void test1() {  
        Syso("Hello ")
```

```
    }
```

```
Class B extends ClassA{
```

```
    Public void test2(){  
        Syso("Hello java");
```

```
    Public static void main(String [] args){  
        ClassB obj = new ClassB();  
        obj.test1();  
        obj.test2();  
    }
```

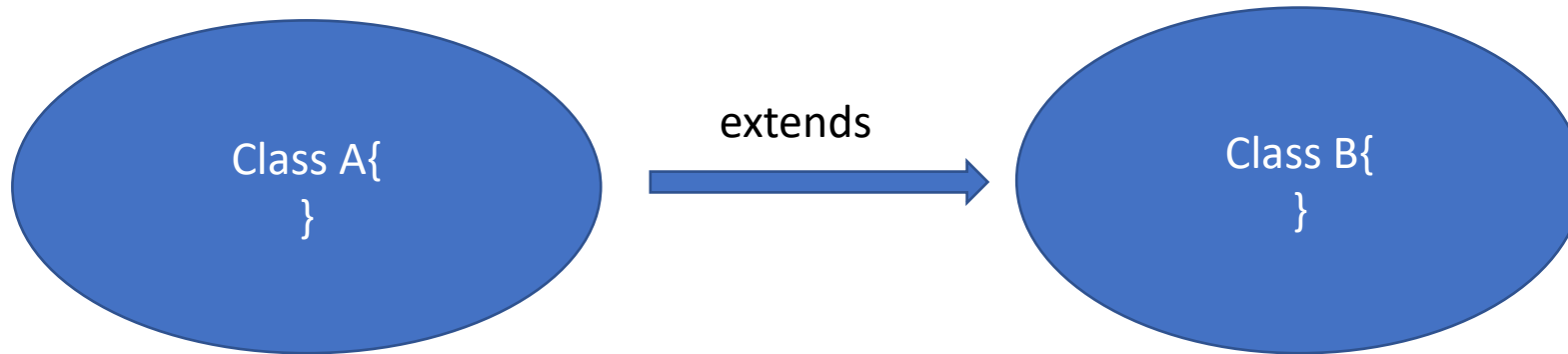


Here we see that test1 was method of ClassA but we have called it with the object reference of ClassB because Class B have inherited all the property of ClassA now

Types of inheritance:-

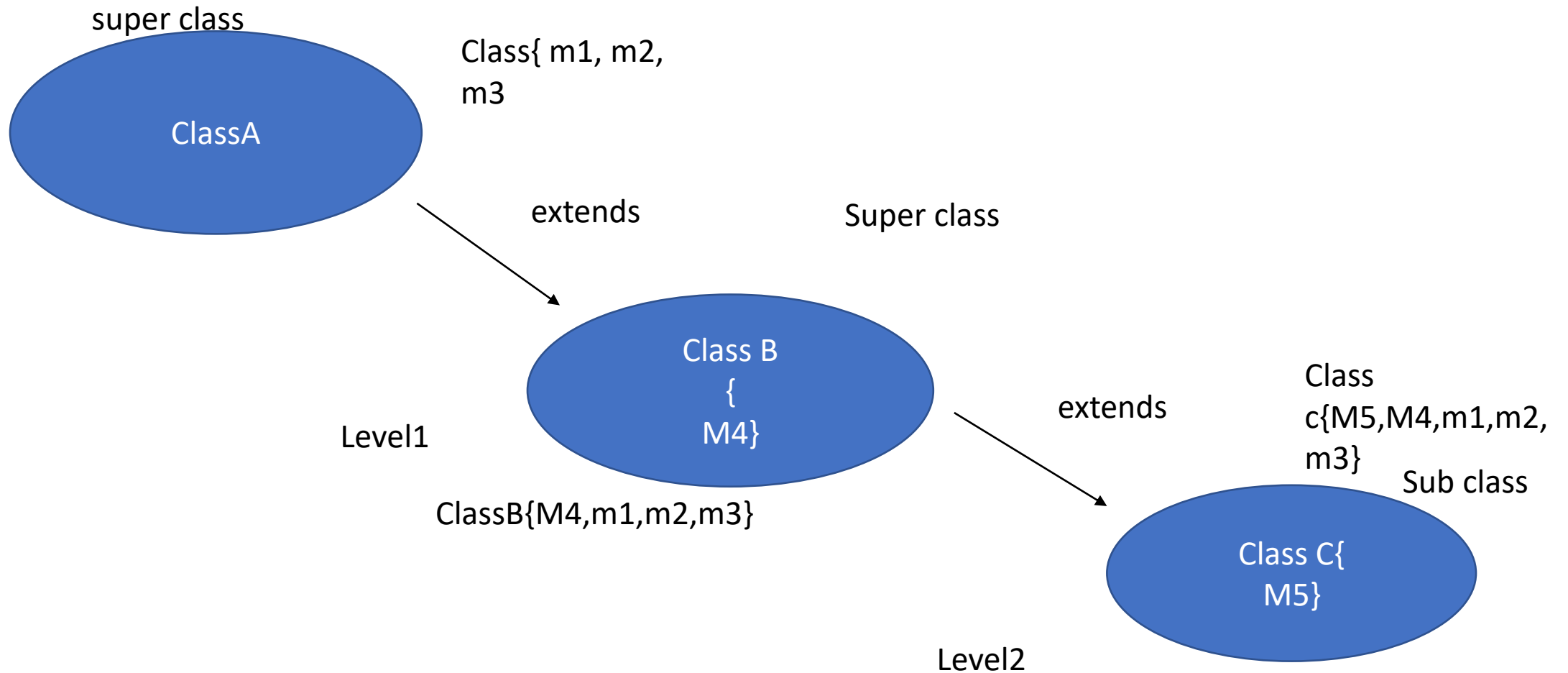
- Single Level Inheritance.
- Multi Level Inheritance.
- Multiple Inheritance.
- Hierarchical Inheritance/Hybrid Inheritance.

1.Single Level Inheritance :- When one class acquires the property of another class, or when a single sub class inherits the property of a single super class this is known as Single level inheritance.



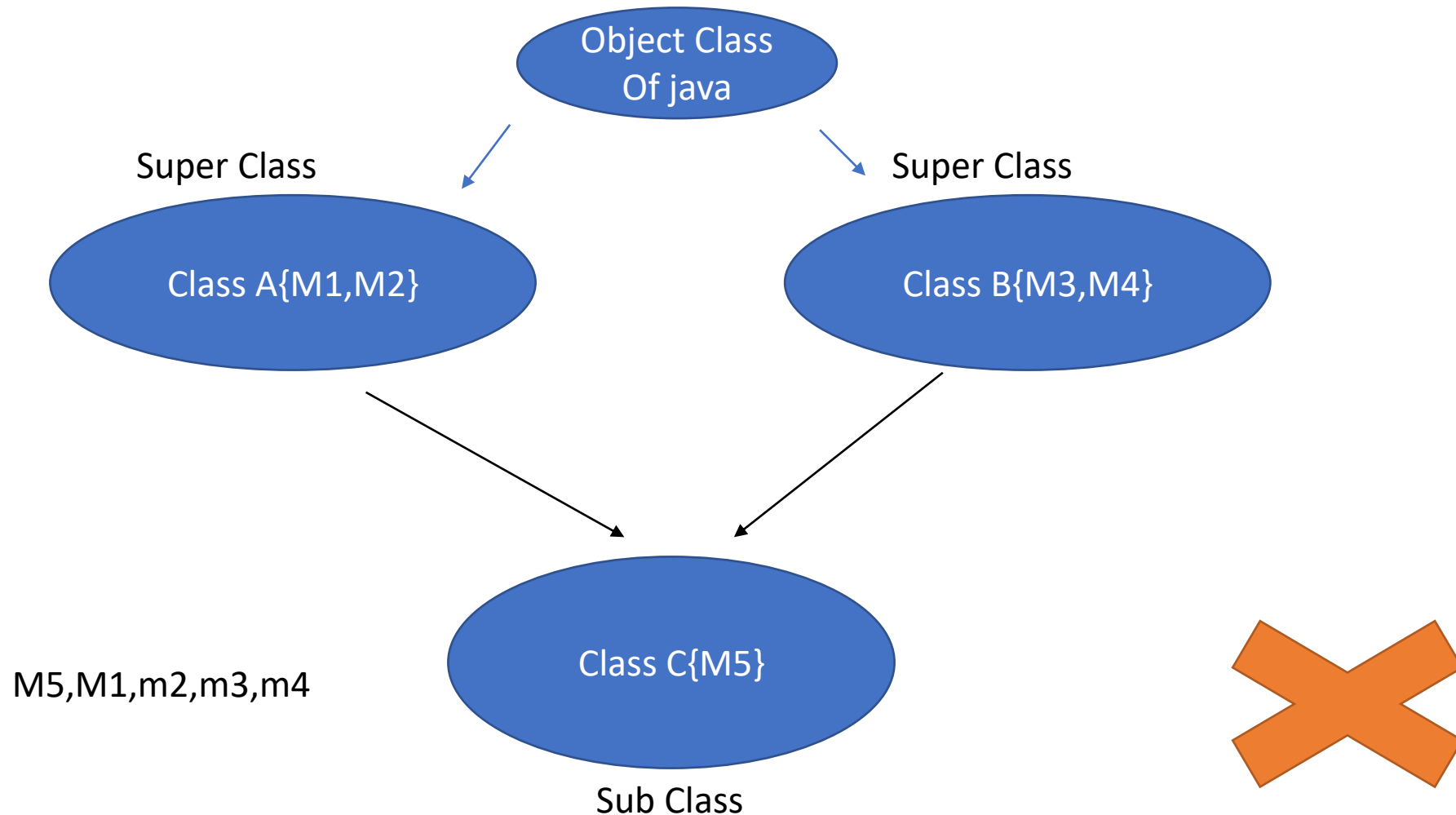
## 2.Multilevel Inheritance:-

When a subclass acquires the property of superclass and then this subclass is again inherited by another subclass is known as multilevel Inheritance.



### 3. Multiple Inheritance : -

when one subclass acquires the property of multiple super class its called as multiple inheritance.



Multiple inheritance is not possible for class in Java due to diamond ambiguity problem.

Object is the super most class of java so in this case java gets confused that which route it has to take to call the supermost object as Subclass points towards two super class



#### 4. Hierarchical :-

When multiple sub class extends one super class, i.e. when more than one sub class inherits the property of single super class , this is known as Hierarchical/Hybrid Inheritance.

