



track of IT

215/318, Hasina Monjil (1st Floor), Muradpur, Panchlaish, Chittagong.
Website: www.rcreation-bd.com, Contact: +8801722-964303

Lecture – 8 (Inheritance)

Inheritance in Java

Inheritance in java is a mechanism in which one object acquires all the properties and behaviors of parent object.

The idea behind inheritance in java is that you can create new classes that are built upon existing classes. When you inherit from an existing class, you can reuse methods and fields of parent class, and you can add new methods and fields also.

Inheritance represents the **IS-A relationship**, also known as parent-child relationship.

Why use inheritance in java

- For Method Overriding (so runtime polymorphism can be achieved).
- For Code Reusability.

Syntax of Java Inheritance

```
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. The meaning of "extends" is to increase the functionality.

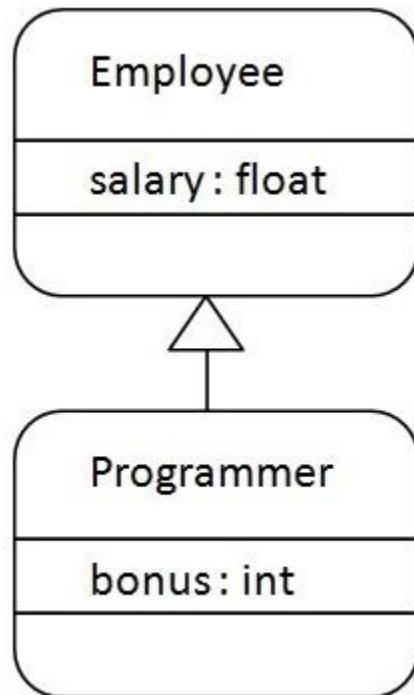
In the terminology of Java, a class which is inherited is called parent or super class and the new class is called child or subclass.

R-Creation

track of IT

215/318, Hasina Monjil (1st Floor), Muradpur, Panchlaish, Chittagong.
Website: www.rcreation-bd.com, Contact: +8801722-964303

Java Inheritance Example



As displayed in the above figure, Programmer is the subclass and Employee is the superclass. Relationship between two classes is **Programmer IS-A Employee**. It means that Programmer is a type of Employee.

```
class Employee{
    float salary=40000;
}
class Programmer extends Employee{
    int bonus=10000;
    public static void main(String args[]){
        Programmer p=new Programmer();
        System.out.println("Programmer salary is:"+p.salary);
        System.out.println("Bonus of Programmer is:"+p.bonus);
    }
}
```

R-Creation

track of IT

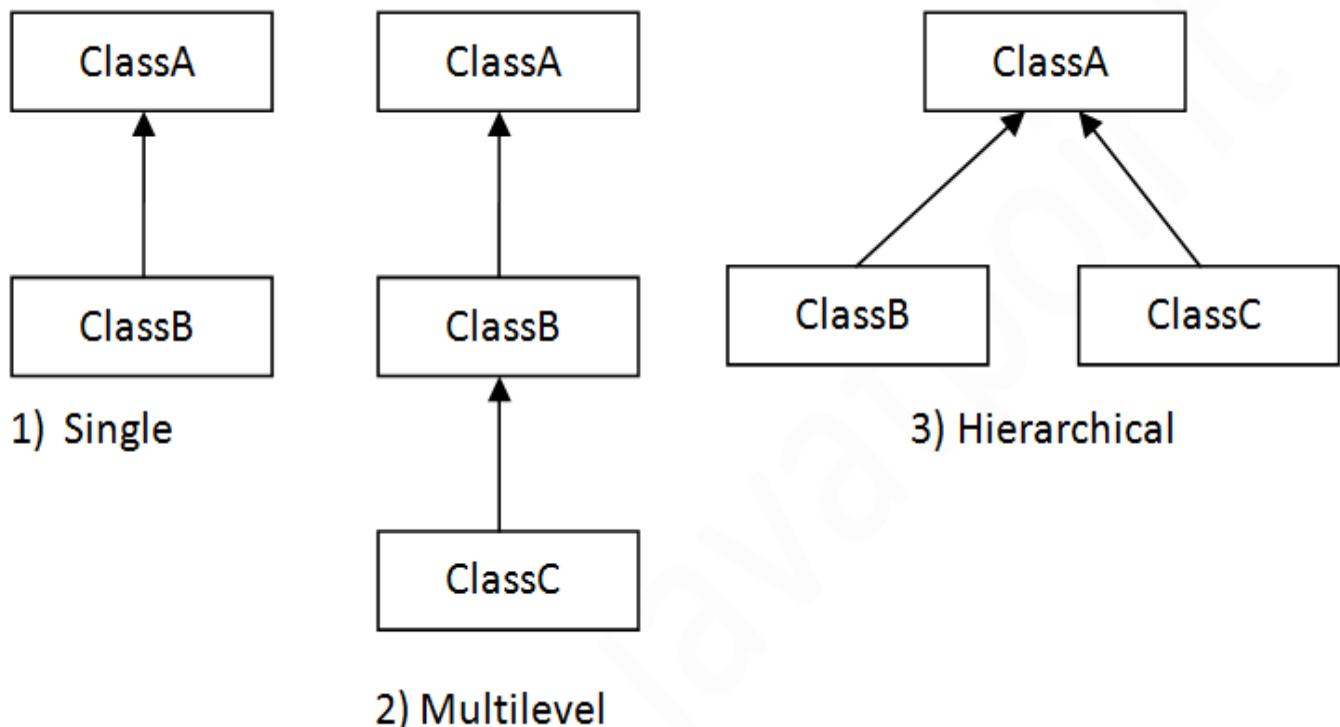
215/318, Hasina Monjil (1st Floor), Muradpur, Panchlaish, Chittagong.
Website: www.rcreation-bd.com, Contact: +8801722-964303

In the above example, Programmer object can access the field of own class as well as of Employee class i.e. code reusability.

Types of inheritance in java

On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.

In java programming, multiple and hybrid inheritance is supported through interface only. We will learn about interfaces later.



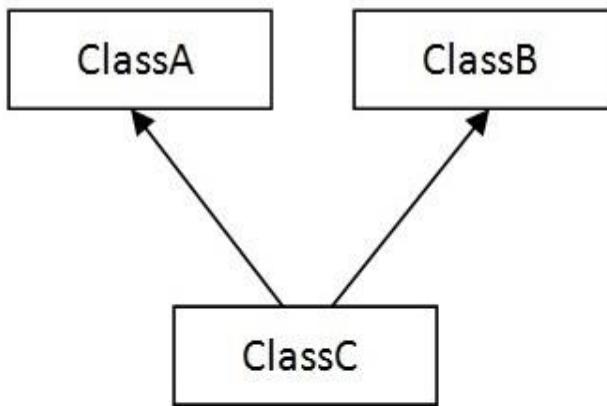
Note: Multiple inheritance is not supported in java through class.

When a class extends multiple classes i.e. known as multiple inheritance. For Example:

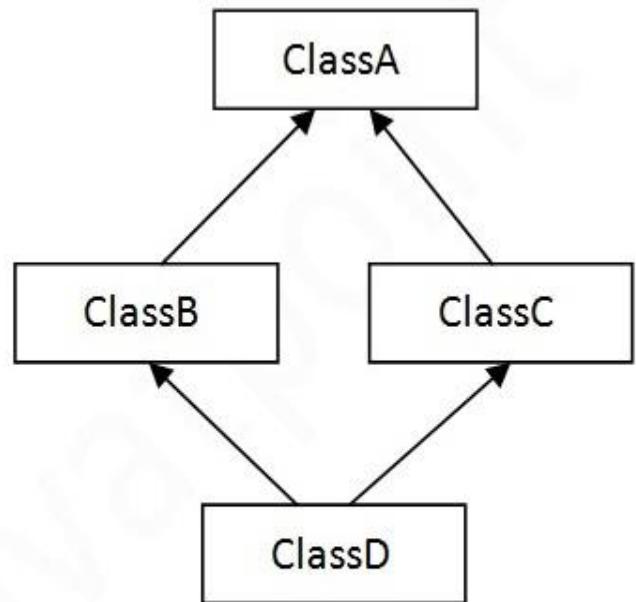
R-Creation

track of IT

215/318, Hasina Monjil (1st Floor), Muradpur, Panchlaish, Chittagong.
Website: www.rcreation-bd.com, Contact: +8801722-964303



4) Multiple



5) Hybrid

Single Inheritance Example

File: TestInheritance.java

```
class Animal{  
    void eat(){  
        System.out.println("eating...");  
    }  
}  
  
class Dog extends Animal{  
    void bark(){  
        System.out.println("barking...");  
    }  
}  
  
class TestInheritance{
```

R-Creation

track of IT

215/318, Hasina Monjil (1st Floor), Muradpur, Panchlaish, Chittagong.
Website: www.rcreation-bd.com, Contact: +8801722-964303

```
public static void main(String args[]){
Dog d=new Dog();
d.bark();
d.eat();
}
}
```

Multilevel Inheritance Example

File: TestInheritance2.java

```
class Animal{
void eat(){
System.out.println("eating... ");
}
}
class Dog extends Animal{
void bark(){
System.out.println("barking... ");
}
}
class BabyDog extends Dog{
void weep(){
System.out.println("weeping... ");
}
}
class TestInheritance2{
public static void main(String args[]){
BabyDog d=new BabyDog();
d.weep();
d.bark();
d.eat();
}
}
```

R-Creation

track of IT

215/318, Hasina Monjil (1st Floor), Muradpur, Panchlaish, Chittagong.
Website: www.rcreation-bd.com, Contact: +8801722-964303

Hierarchical Inheritance Example

File: TestInheritance3.java

```
class Animal{
    void eat(){
        System.out.println("eating... ");
    }
}
class Dog extends Animal{
    void bark(){
        System.out.println("barking... ");
    }
}
class Cat extends Animal{
    void meow(){
        System.out.println("meowing... ");
    }
}
class TestInheritance3{
    public static void main(String args[]){
        Cat c=new Cat();
        c.meow();
        c.eat();
        //c.bark(); //C.T.Error
    }
}
```

Q) Why multiple inheritance is not supported in java?

To reduce the complexity and simplify the language, multiple inheritance is not supported in java.

Consider a scenario where A, B and C are three classes. The C class inherits A and B classes. If A and B classes have same method and you call it from child class object, there will be ambiguity to call method of A or B class.

R-Creation

track of IT

215/318, Hasina Monjil (1st Floor), Muradpur, Panchlaish, Chittagong.
Website: www.rcreation-bd.com, Contact: +8801722-964303

Since compile time errors are better than runtime errors, java renders compile time error if you inherit 2 classes. So whether you have same method or different, there will be compile time error now.

```
class A{
void msg(){
System.out.println("Hello");
}
}
class B{
void msg(){
System.out.println("Welcome");
}
}
class C extends A,B{//suppose if it were

Public Static void main(String args[]){
C obj=new C();
obj.msg(); //Now which msg() method would be invoked?
}
}
```

Compile Time Error