Course Name - Object Oriented Programming using Java

Lecture 19: Inheritance - Definition, Advantages, Different Types of Inheritance and their Implementation.

Presented By
Dr. Sudipta Sahana
Associate Professor
Dept. of CSE
UEM - Kolkata

Topic of Interest

- **Definition and Advantages of Inheritance**
- **Single Inheritance**
- **Single Inheritance Example**
- **Hierarchical Inheritance**
- **Hierarchical Inheritance Example**
- **Multilevel Inheritance**
- **Multilevel Inheritance Example**
- **Multiple Inheritance**
- **Why multiple inheritance is not supported in java?**

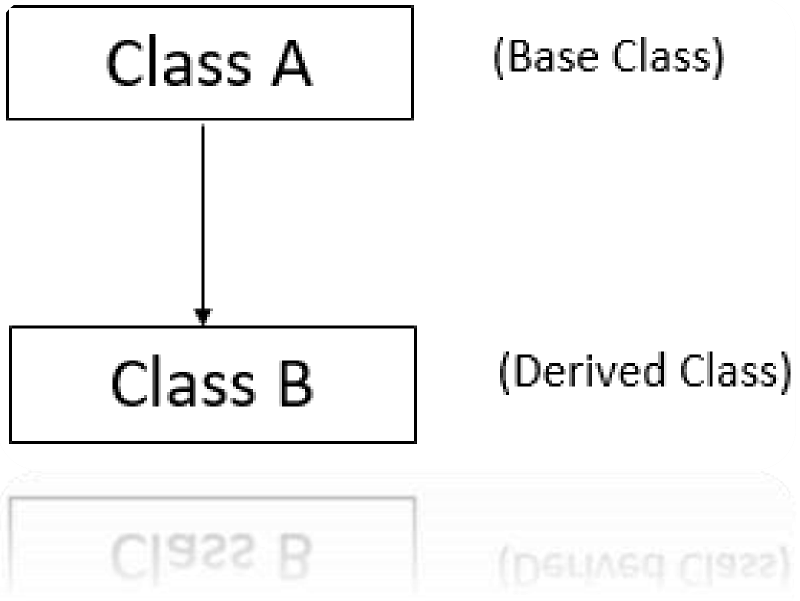
Definition and Advantages of Inheritance

Definition of Inheritance: Inheritance is a process through which some physical appearance (Data) and some logical behavior (method) related properties of parent class can be acquired by it’s child class without any modification.

Advantages of inheritance:
In case of inheritance a piece of code (Data as well as method) can be introduced only once in parent class that can be automatically inherited into its child class and Code is reused.

Single Inheritance

Single inheritance is damn easy to understand. When a class extends another one class only then we call it a single inheritance. The below flow diagram shows that class B extends only one class which is A. Here A is a **parent class** of B and B would be a **child class** of A.



Single Inheritance Example

```
class Employee
 {  int  ID_No;
      void Salary_draw(){
              System.out.println(“Every employee
must draw their salary”);
 }}
class Teaching_Staff  extends Employee
 {void Teach() {
              System.out.println(“Every teacher can
teach their students”);
 }
}
```

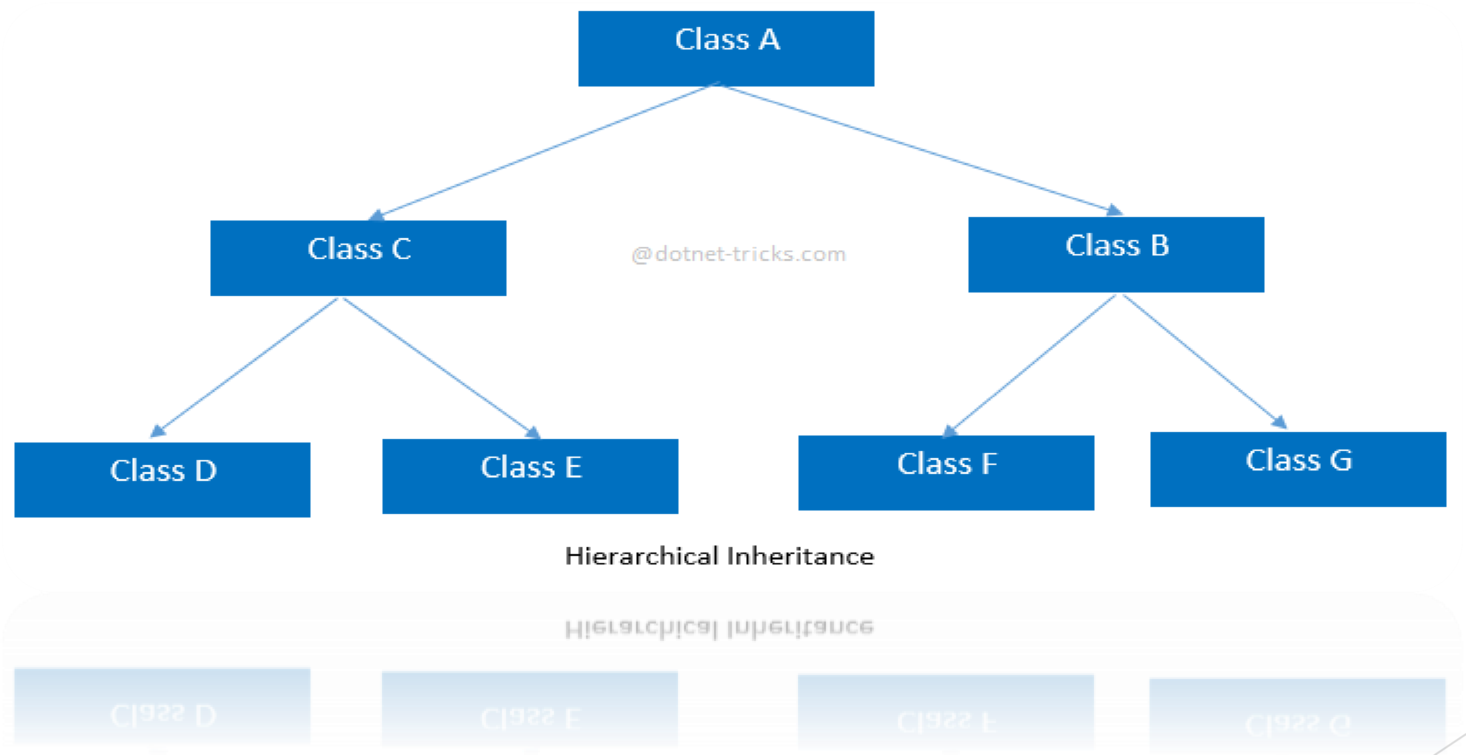
```
class Inherit
 {  public static void main(String[]  args)
      {
  Teaching_Staff  SM = new Teaching_Staff ();
         SM.ID_No=142;
         SM.Salary_draw();
         SM. Teach();
         System.out.println(“ID:=”+ SM.ID_No);
          }
}
```

OUTPUT:

```
Every employee must draw their
salary
Every teacher can teach their students
ID:=142
```

Hierarchical Inheritance

In this inheritance One parent and more than one child class exists in two different but consecutive levels, and properties acquired from parent class to all child class that exists in same level without any modification..



Hierarchical Inheritance

Hierarchical Inheritance Example

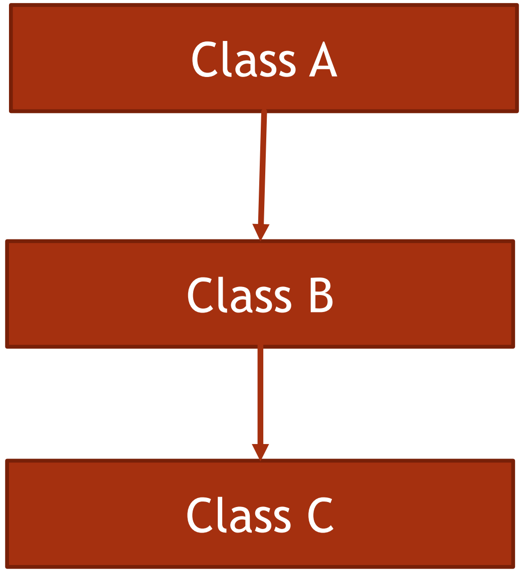
```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class Cat extends Animal{
void meow(){System.out.println("meowing...");}
}
class TestInheritance3{
public static void main(String args[]){
Cat c=new Cat();
c.meow();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

Output:

```
meowing...
eating...
```

Multilevel Inheritance:

In this inheritance one parent single child relationship exists in more than one consecutive levels in chain form, and properties acquired from grand parent class to parent and from parent class into it’s child class without any modification.



Multilevel Inheritance Example

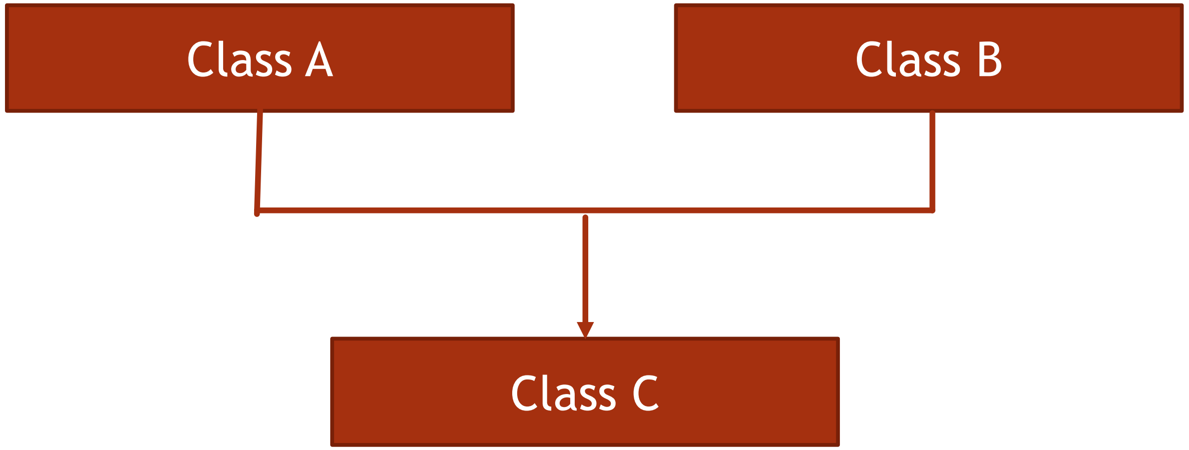
```
class Animal{
void eat(){System.out.println("eating...");}
}
class Dog extends Animal{
void bark(){System.out.println("barking...");}
}
class BabyDog extends Dog{
void weep(){System.out.println("weeping...");}
}
class TestInheritance2{
public static void main(String args[]){
BabyDog d=new BabyDog();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

Output:

```
weeping...
barking...
eating...
```

Multiple Inheritance

In this inheritance multiple no of parents exists in single level. Properties of those parent classes can be acquired into the single child class which resided in the recent next level without any modification. Java does not support multiple inheritance. In interface section I can elaborately discuss Multiple Inheritance.



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 the same method and you call it from child class object, there will be ambiguity to call the method of A or B class. 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.

```
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 i
nvoked?
}
}
```

Output
Compile Time Error

Thank You