



Object-Oriented Programming (CS F213)

Module III: Inheritance and Polymorphism in Java

CS F213 RL 9.1: Inheritance Basics-I

BITS Pilani

Dr. Pankaj Vyas

Department of Computer Science, BITS-Pilani, Pilani Campus

CS F213 RL 9.1 : Topics



- Inheritance in Java

Inheritance Basics

- Reusability is achieved by *Inheritance*
- Java classes can be Reused by Extending a class. Extending an existing class is nothing but reusing properties of the existing classes.
- The class whose properties are extended is known as *super or base or parent class*.
- The class which extends the properties of super class is known as *sub or derived or child class*
- A class can either **extends another class** or **can implement an interface**

Java's Support For Inheritance



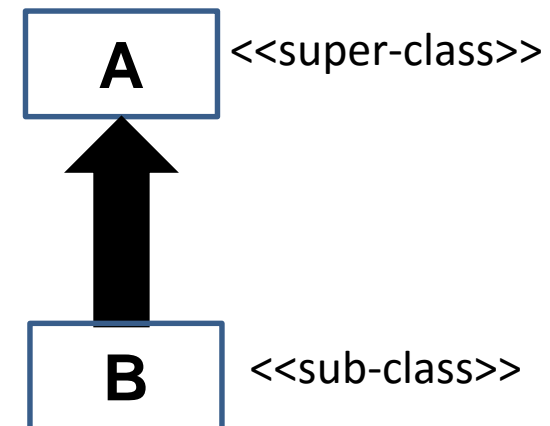
- By Extending a Class

- Syntax

```
class      class-name      extends      super-class-name
{
    .....
}
```

- Example

```
class A
{
    .....
} // End of class A
class B extends A
{
    .....
} // End of class B
```



Java's Support For Inheritance



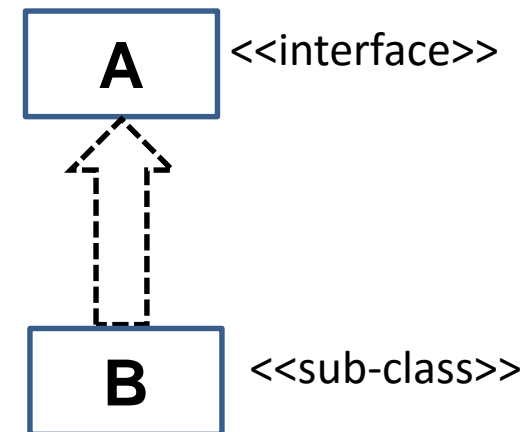
- By Implementing an Interface

- Syntax

```
class      class-name      implements      interface-name
{
    .....
}
```

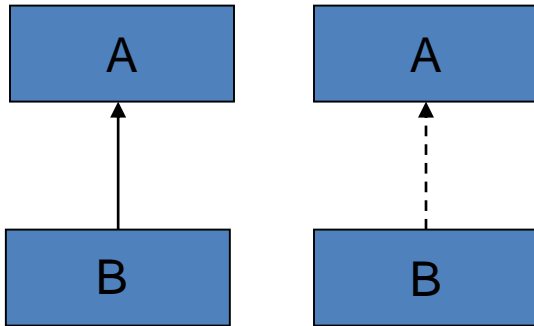
- Example

```
interface A
{
    .....
} // End of interface A
class B implements A
{
    .....
} // End of class B
```

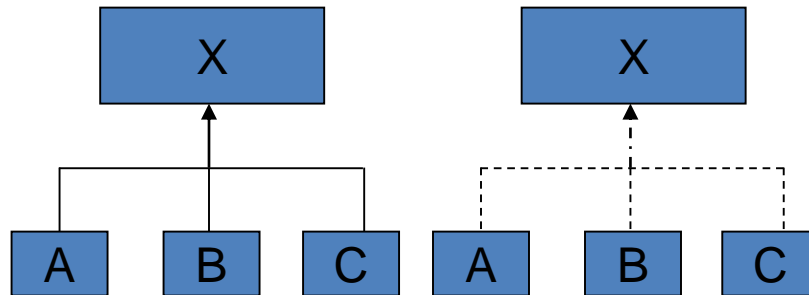


Forms of Inheritance

Single Inheritance



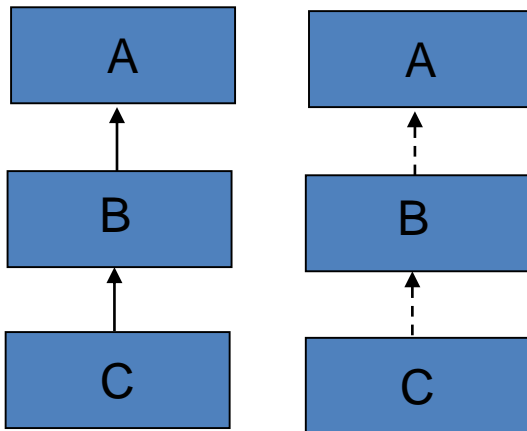
Hierarchical Inheritance



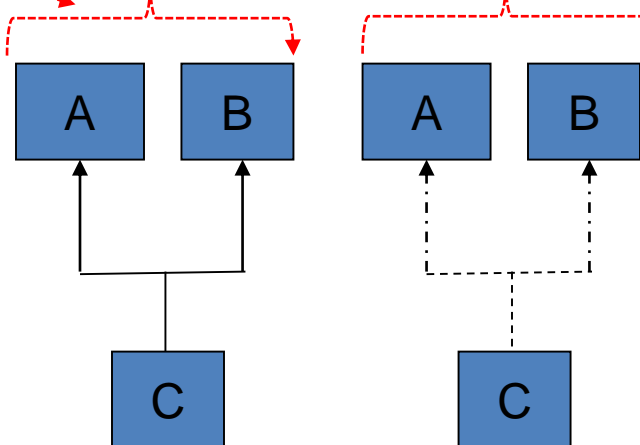
**Multiple Class Inheritance
NOT SUPPORTED BY JAVA**

**Multiple Interface Implementation
SUPPORTED BY JAVA**

Multi-Level Inheritance



Multiple Inheritance



Multiple Inheritance in Java



- Java Does Not Support Multiple Inheritance by Extending Multiple Classes
- The Following Code is Wrong

```
class A      {      } // End of class A
class B      {      } // End of class B
class C      extends A, B
{
} // End of class C
```

**Wrong Java Does not
Allow Multiple Class
Extensions**

Multiple Inheritance in Java



- Java Supports Multiple Inheritance by Implementing Multiple Interfaces
- The Following Code is Correct

```
interface A {      } // End of interface A
interface B {      } // End of interface B
```

```
class C      implements A, B
{
} // End of class C
```

**Java Allows Multiple
Interface
Implementations**

Thank You