



OOPs Concepts



Object Oriented Programming System

- Object means a real word entity such as pen, chair, table etc.
- Object-Oriented Programming is a methodology or paradigm to design a program using classes and objects.

Class

- *Collection of objects is called class.*

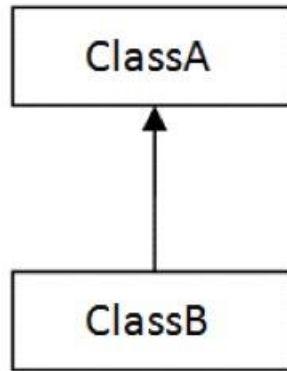
Example:

```
public class Print {  
    public static void main(String[] args) {  
    }  
}  
  
class student{  
    String name;  
    int standard;  
    int rollno;  
}
```

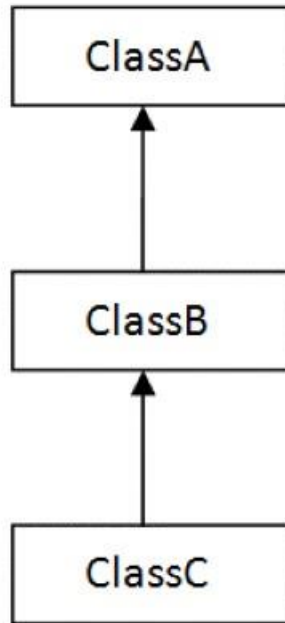
Inheritance

- When one object acquires all the properties and behaviors of parent object i.e. known as inheritance.
- It provides code reusability.

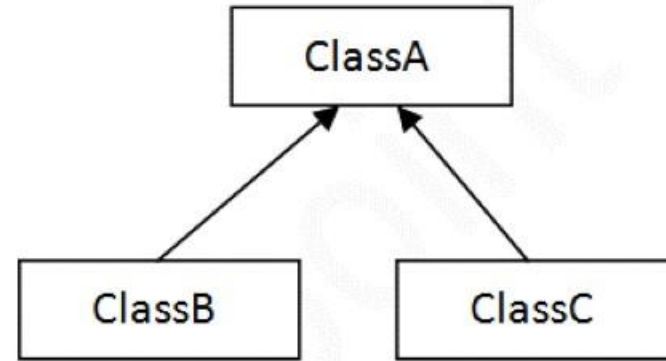
Types of inheritance



1) Single



2) Multilevel



3) Hierarchical

Single Inheritance Example

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

Output: barking...
eating...


```
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();
    }
}
```

Multilevel Inheritance Example

Output: weeping...
barking
eating

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...