Contents

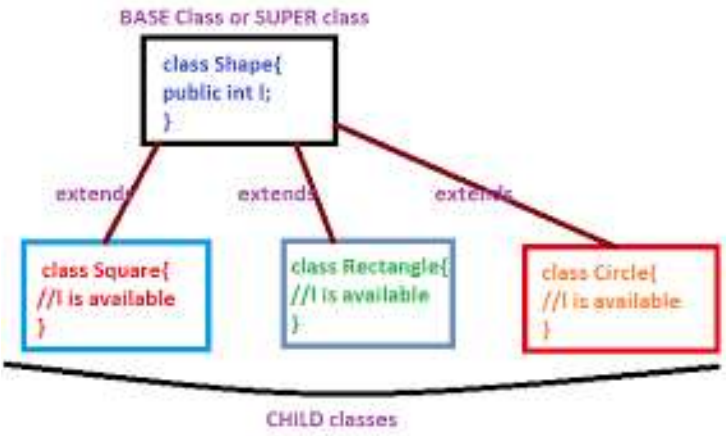
Inheritance & Polymorphism

Inheritance:

- Child class (sub class) automatically inherits the variables and methods defined in its parent class (super class).
- This allows to reuse existing class members and make them extendible either for enhancement or alteration.

Extends keyword :

class Base { //code }
class Derived **extends** Base { //code }

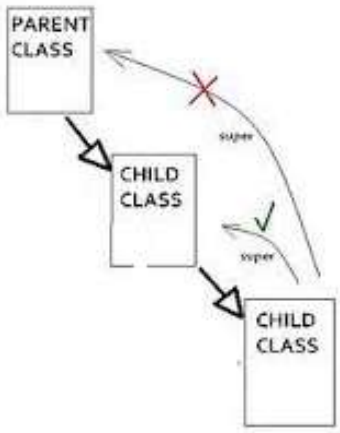


super

It is a reference variable used to refer the immediate parent class object

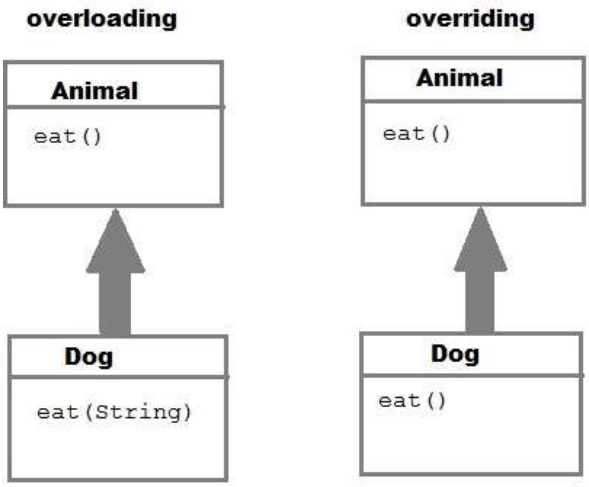
- super() invokes immediate parent class constructor
- Call member (variables & methods) of parent class

Syntax : super.baseclassMemberName



Overloading & Overriding

Overloading (compile time polymorphism)	Overriding (run time polymorphism)
Two or more methods within the same class share the same name but parameter declarations are different. You can overload Constructors and Normal Methods.	A method in a subclass has the same name and type signature as a method in its super class, then the subclass method overrides the super class method. Overridden methods allow Java to support run-time polymorphism



Overriding

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }
}
class Hound extends Dog{
    public void sniff(){
        System.out.println("sniff ");
    }

    public void bark(){
        System.out.println("bowl");
    }
}
```

Same Method Name, Same parameter

Overloading

```
class Dog{
    public void bark(){
        System.out.println("woof ");
    }

    //overloading method
    public void bark(int num){
        for(int i=0; i<num; i++)
            System.out.println("woof ");
    }
}
```

Same Method Name, Different Parameter

final

- Restrict changing value of variables

- Restricts inheritance. Therefore, restricts method overriding