

Java for beginners

Overloading and Overriding



Saurabh Shukla (MySirG)

Agenda

- ① Revise function overloading in C++
- ② Method Overloading in Java
- ③ Method Overriding in C++
- ④ Method Overriding in Java
- ⑤ Method Hiding

Function Overloading

C++

Two functions with the same name
but differ in arguments.

Overloaded versions must resides
in a single scope ..

C++

class A
{
public:
 void f1() { ... }
};

B obj;
obj.f1();
obj.f1(5);

};
class B : public A
{
public:
 void f1(int);
};

Function prototype

return type functionName (arguments)

①

②

③

Function Signature

functionName (arguments)

①

②

Function Name

functionName

④

Method Overloading

Methods sharing same name but different signatures, is known as Overloaded methods.

Overloaded methods provide similar services but different implementations

Unlike C++, Java truly extends
the scope of class in inheritance.

Method Overloading is possible even
if the one version resides in
superclass and another version
resides in subclass.

Method Overloading is a way to
implement the concept of
Polymorphism.

Class A

{

void show() { }

void show(int) { }

void show(int, int, int) { }

}

Class A

{

void show(int) { }

void show(String) { }

}

class A

{

void show(int, string) { }

void show(string, int) { }

}

Class A

{

void show(int) { }

float show(int) { }

}

No
~~Same~~ Same Signature
because

Java

```
class A  
{  
    public void f1() {  
        =  
    }  
}
```

This is
method overload
in java

obj.f1() ✓
obj.f1(5) ✓

```
class B extends A  
{  
    public void f1(int x){  
        --  
    }  
}
```

Method Overriding

C++

Functions with the same prototype but different implementations, one reside in superclass and another in subclass, is known as Function Overriding.

C++

Class A

{ public:

void f1 (int a) { ... }

};

class B : public A

{

public:

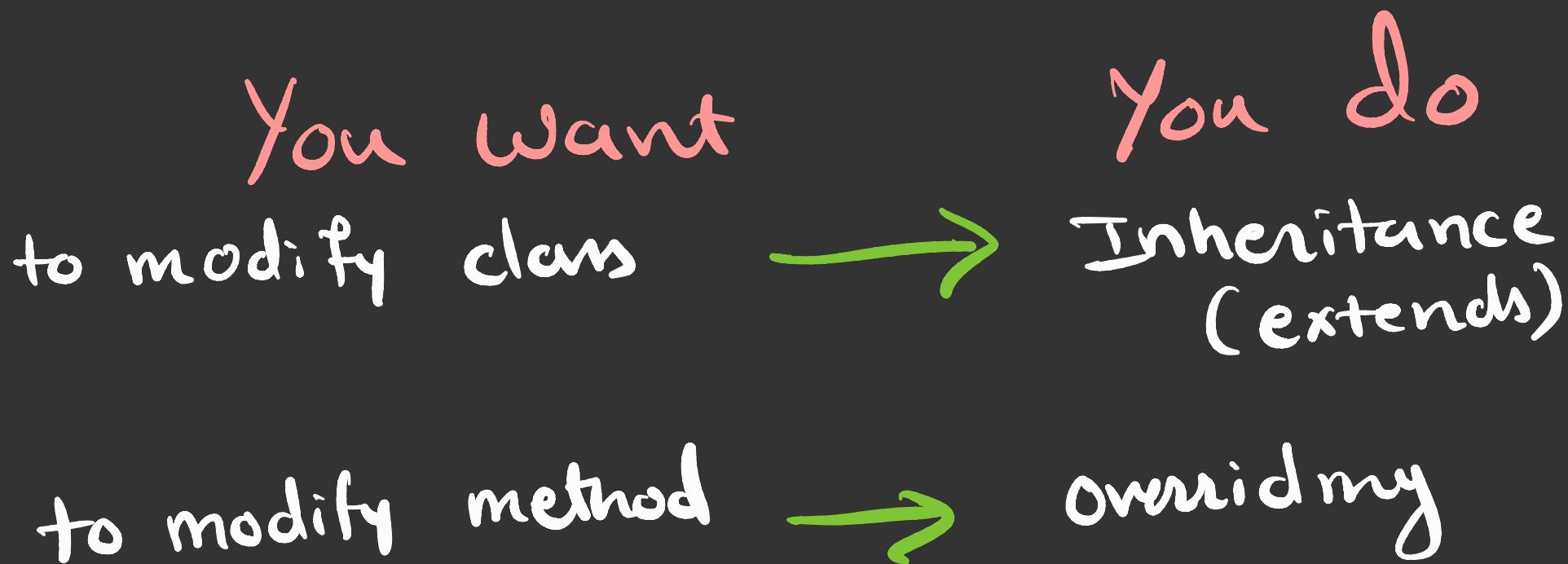
void f1 (int a) { ... }

};

function
overriding

Why Overriding ?

When subclass wants to modify superclass method for its user, subclass has to override the superclass method in the subclass.



Whenever you need to modify Superclass
method definition with respect to subclass,
you have to override the method.

C++

Virtual Keyword in C++

Class A

{

public:

virtual void f1() {..}

B obj:

obj.f1(); //B

A *P = new B;

P->f1(); //A B

};

Class B : public A

{

public:

void f1() {..}

};

P = new A;

P->f1(); //A

Method Overriding in Java

Overridden methods in Java are binded dynamically, so there is no need of virtual function/Keyword.

```
B b1 = new B();
b1.f1();
```

In Java, access specifier of overridden method in the subclass can be same as of method in Superclass or lesser restrictive.

Java

class A
{

 void f1() { .. }

private
default
protected
public

{
class B extends A
{

 public void f1() { .. }

}

Method Hidng

static method with same prototype
one in Super class and another in
Sub class is method hidng.

Class A

{

 static void sf1() { .. }

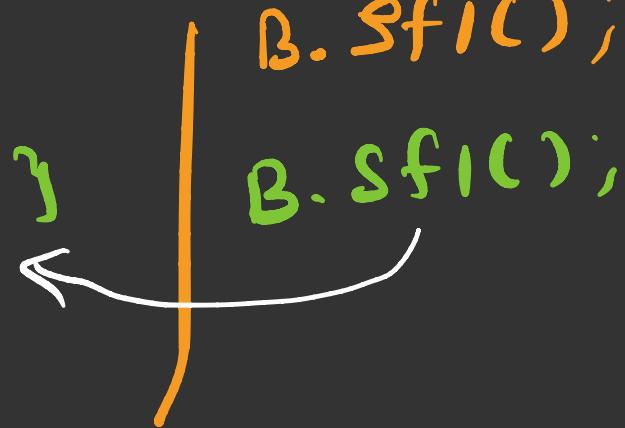


}

class B extends A

{

 static void sf1() { .. }



}