

# Difference Between this and super in Java

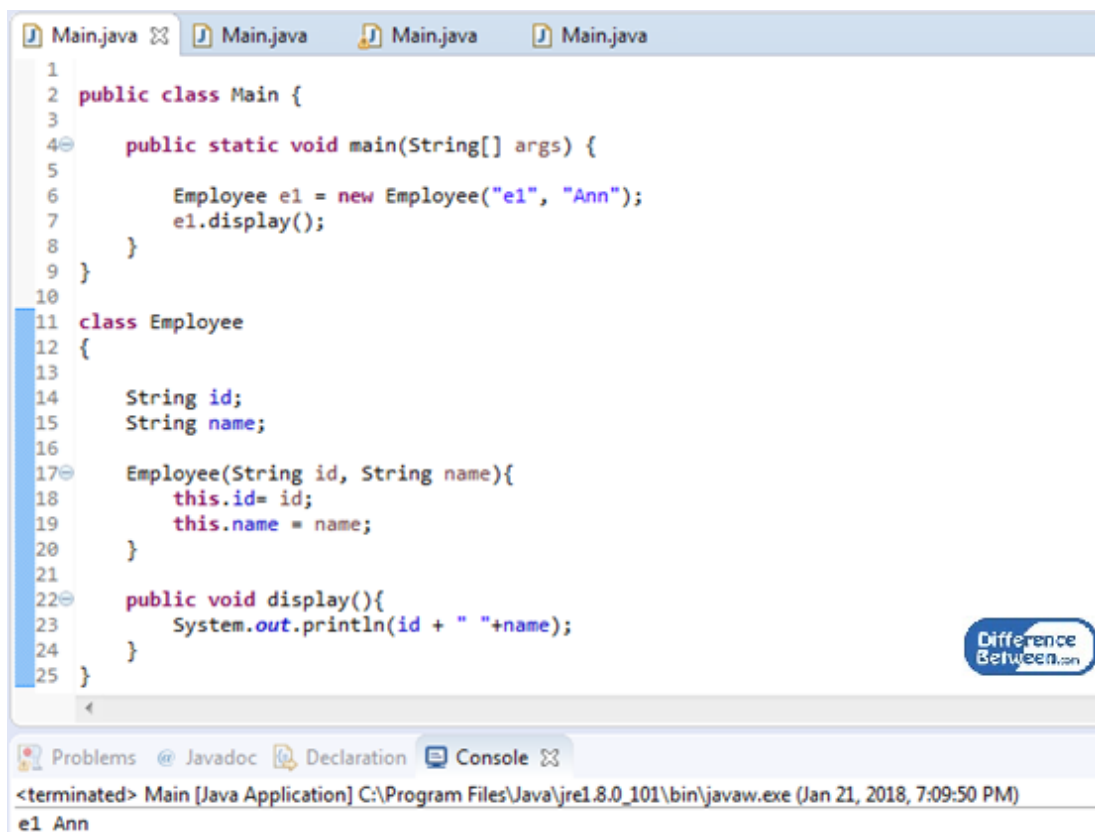
[www.differencebetween.com](http://www.differencebetween.com)

## Key Difference - this vs super in Java

The keywords 'this' and 'super' are used in [Java](#) programming. These keywords cannot be used as variables or any other identifier name. Java supports [Object Oriented Programming](#) (OOP). The program or [software](#) can be modeled using [objects](#). Objects are insatiate using classes. One pillar of OOP is [inheritance](#). It provides code reusability. The classes that already exist are [superclasses](#), and the derived classes are [subclasses](#). The super keyword can be used to refer an object of the superclass. There are multiple objects in the system. The 'this' keyword is used to refer a current object. The **key difference** between this and super is **'this' is a reference variable that is used to refer the current object while 'super' is a reference variable that is used to refer immediate superclass object.**

## What is this in Java?

The keyword 'this' is used to refer a current object. Refer the given Java program.



```
1
2 public class Main {
3
4     public static void main(String[] args) {
5
6         Employee e1 = new Employee("e1", "Ann");
7         e1.display();
8     }
9 }
10
11 class Employee
12 {
13
14     String id;
15     String name;
16
17     Employee(String id, String name){
18         this.id= id;
19         this.name = name;
20     }
21
22     public void display(){
23         System.out.println(id + " "+name);
24     }
25 }
```

<terminated> Main [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jan 21, 2018, 7:09:50 PM)  
e1 Ann

Figure 01: Java program using this keyword

In Java, there are three types of variables. They are instance variables, local variables and class variables. According to the above program, class Employee has two instance variables. They are id and name. Local variables are the variables belongs to methods. Class variables are shared by all objects. The id and name are passed to the Employee constructor. If the programmer writes id = id; it won't initialize the instance variables because the Constructor already has id and name. There are no values for instance variables. So, printing them will display null. When using this, it refers to the current object. Therefore, giving id and name to the constructor can set the instance variables.

The keyword 'this' can be used to invoke the current class method. Refer to Java program.

```
public class ThisDemo{  
  
    public static void main(String[] args){  
  
        Myclass myClass= new Myclass();  
  
        myClass.B();  
  
    }  
  
}  
  
class Myclass{  
  
    public void A(){  
  
        System.out.println("A");  
  
    }  
  
    public void B(){  
  
        System.out.println("B");  
  
        this.A();  
  
    }  
  
}
```

The class Myclass contains two methods. They are method A and B. When creating an object of Myclass and invoking the method B will print B, A as the output. In method B, after printing B there is a statement as this.A(). Using this, the current class method was invoked.

It is also possible to use this keyword to invoke the current class constructor. Refer the given program.

```
public class ThisDemo{

public static void main(String[] args){

A obj = new A(5);

}

}

class A{

public A(){

System.out.println("Constructor A");

}

public A(int x){

this();

System.out.println("Parameterized Constructor A");

}

}
```

According to the above program, class A has a default constructor and a parameterized constructor. When creating an object of A, the parameterized constructor is called. In the parameterized constructor, there is a statement like `this()`; It will call the current class constructor that is `A()`.

## **What is super in Java?**

The keyword 'super' is related to inheritance. Inheritance is a major concept of Object Oriented Programming. It allows using the properties and methods of the already existing class to a new class. The already existing class is known as the parent class or superclass. The new class is known as the child class or subclass.

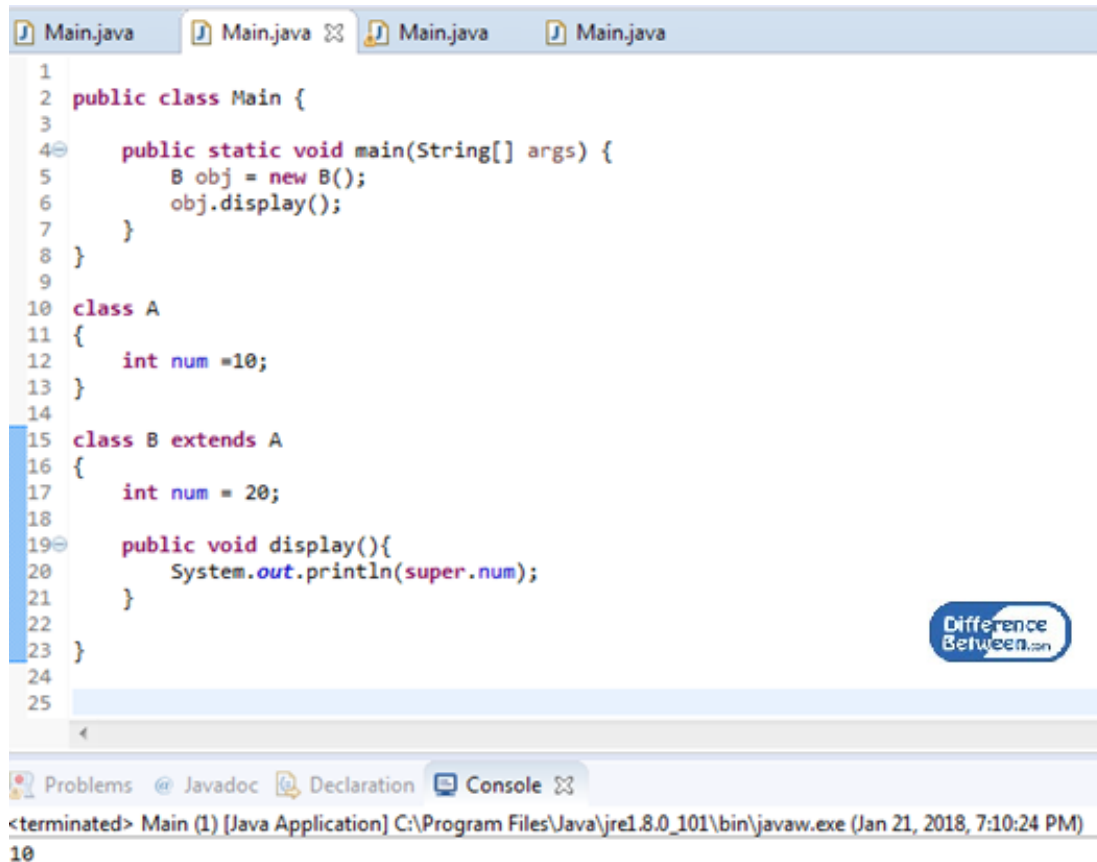
The 'super' is a reference variable that is used to refer the immediate parent class object. The super keyword can refer immediate parent class instance variable or invoke

immediate parent class method. The `super()` is used to invoke immediate parent class constructor.

Assume that there are two classes as A and B. Class A is the superclass and class B is the subclass. Class A, B both have display method.

```
public class A{  
  
    public void display(){  
  
        System.out.println("A");  
  
    }  
  
}  
  
public class B extends A{  
  
    public void display(){  
  
        System.out.println("B");  
  
    }  
  
}
```

When creating an object of type B and calling the method `display`, it will give the output B. class A has the `display` method, but it is overridden by the subclass B `display` method. If the programmer wants to call the `display` method in class A, then he can use the `super` keyword. Refer the given Java program.



```
1
2 public class Main {
3
4     public static void main(String[] args) {
5         B obj = new B();
6         obj.display();
7     }
8 }
9
10 class A
11 {
12     int num =10;
13 }
14
15 class B extends A
16 {
17     int num = 20;
18
19     public void display(){
20         System.out.println(super.num);
21     }
22 }
23
24
25
```

Problems Javadoc Declaration Console

<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jan 21, 2018, 7:10:24 PM)

10

**Figure 02: Java program using super keyword**

According to the above program, class A has a variable named number with value 10. Class B extends A and has a variable named number with value 20. Generally, when creating an object of type B and calling the display method should give the number in the subclass because the superclass value is overridden by the new class. By using the super.num, the superclass number value is printed.

The super() can be used to call the superclass constructor. Refer the below program.

```
public class Main {

public static void main(String[] args){

B obj = new B();

}

}

class A{
```

```
A(){  
    System.out.println("A");  
}  
  
class B extends A{  
    B(){  
        super();  
        System.out.println("B");  
    }  
}
```

According to the above program, class A has a constructor A (). Class B has the constructor B (). Class B extends class A. When creating an object of type B, it will print A, B as the output. The B () constructor has super (). Therefore, first the A constructor is invoked and then goes to B. Even though super () is not written, by default the parent constructor is called.

The super using the method is as follows.

```

1 public class Main {
2
3
4 public static void main(String[] args) {
5
6     B obj = new B();
7     obj.display();
8 }
9 }
10
11 class A
12 {
13 public void display(){
14     System.out.println("A");
15 }
16 }
17
18 class B extends A
19 {
20 public void display(){
21     super.display();
22     System.out.println("B");
23 }
24 }
25

```

Problems @ Javadoc Declaration Console
   
 <terminated> Main (2) [Java Application] C:\Program Files\Java\jre1.8.0\_101\bin\javaw.exe (Jan 21, 2018, 7:10:46 PM)
   
 A
   
 B

**Figure 03: Java program that invokes the superclass method**

According to the above program, class A has display method. Class B also has display method. Class B extends A. When creating an object of type B and calling the display method will give output as A and B. In class B display method, class A display method is called using `super.display()`. That method prints "A" first. Then prints "B".

## What is the Similarity Between this and super?

- Both are keywords in Java programming.

## What is the Difference Between this and super?

this vs super	
The 'this' is a reference variable that is used to refer the current object.	The 'super' is a reference variable that is used to refer an immediate superclass object.
Instance Variable	

A current class instance variable can be referred using this.	Superclass instance variable can be referred using super.
<b>Class Method</b>	
The current class method can be invoked using this.	Superclass method can be invoked using super.
<b>Constructor</b>	
The current class constructor can be invoked using this().	Superclass constructor can be invoked using super().

## Summary - this vs super in Java

The keywords 'this' and 'super' are used in Java. The keywords cannot be used as variables or any other identifier name. They seem to be the same, but they have a difference. The difference between this and super is that super is a reference variable that is used to refer immediate superclass object while this is a reference variable that refers the current object.

### Reference:

- 1.This() and super keyword in java and its purpose with a short example, Atoz knowledge, 31 Jan. 2015. [Available here](#)
- 2.“This keyword in java - java point.” [Available here](#)
- 3.tutorialspoint.com. “Java Inheritance.” [The Point](#), [Available here](#)

### How to Cite this Article?

APA: Difference Between this and super in Java.(2018 January 29). Retrieved (date), from <http://differencebetween.com/difference-between-this-and-vs-super-in-java/>

MLA: "Difference Between this and super in Java" Difference Between.Com. 29 January 2018. Web.

Chicago: “Difference Between this and super in Java.” Difference Between.Com. <http://differencebetween.com/difference-between-this-and-vs-super-in-java/> accessed (accessed [date]).





