

# SUPER KEYWORD

use of super keyword in Java Hindi [www.smartprogramming.in](http://www.smartprogramming.in)

## Uses of “super” keyword

1. “super” keyword can be used to refer immediate parent class instance variable.
2. “super” keyword can be used to invoke immediate parent class method.
3. super() can be used to invoke immediate parent class constructor.

0:41 / 12:21

super keyword in java Hindi

www.smartprogramming.in

extends B

The diagram illustrates the relationship between a subclass and its superclass. It shows two circles representing objects. The top circle is labeled 'Object A' and contains the letter 'A'. An arrow points from a circle labeled 'this' (with 'reference variable' written below it) to 'Object A'. The bottom circle is labeled 'Object B' and contains the letter 'B'. An arrow points from a circle labeled 'super' (with 'reference variable' written below it) to 'Object B'.

**"super" keyword is a reference variable which is used to refer immediate parent class object**

2:24 / 11:33

www.smartprogramming.in

```

class A
{
    int a=10;
}
class B extends A
{
    int a=20;
    void show(int a)
    {
        sop(a); //30
        sop(this.a); //20
        sop(super.a); //10
    }
}
    
```

```

    sum ( )
    {
        B ob1= new B();
        ob1.show(30);
    }
    
```

```
1 class A
2 {
3     int i=10;
4 }
5 class B extends A
6 {
7     int i=20;
8     void show(int i)
9     {
10        System.out.println(i);           //30
11        System.out.println(this.i);      //20
12        System.out.println(super.i);     //10
13    }
14    public static void main(String[] args)
15    {
16        B ob=new B();
17        ob.show(30);
18    }
19 }
```

**Use 1 : "super" keyword refers to the immediate parent class instance variable**

Java source file | length: 278 | lines: 19 | Ln: 11 | Col: 32 | Sel: 4 | 1 | Windows (CR LF) | UTF-8 | 11-Mar-19

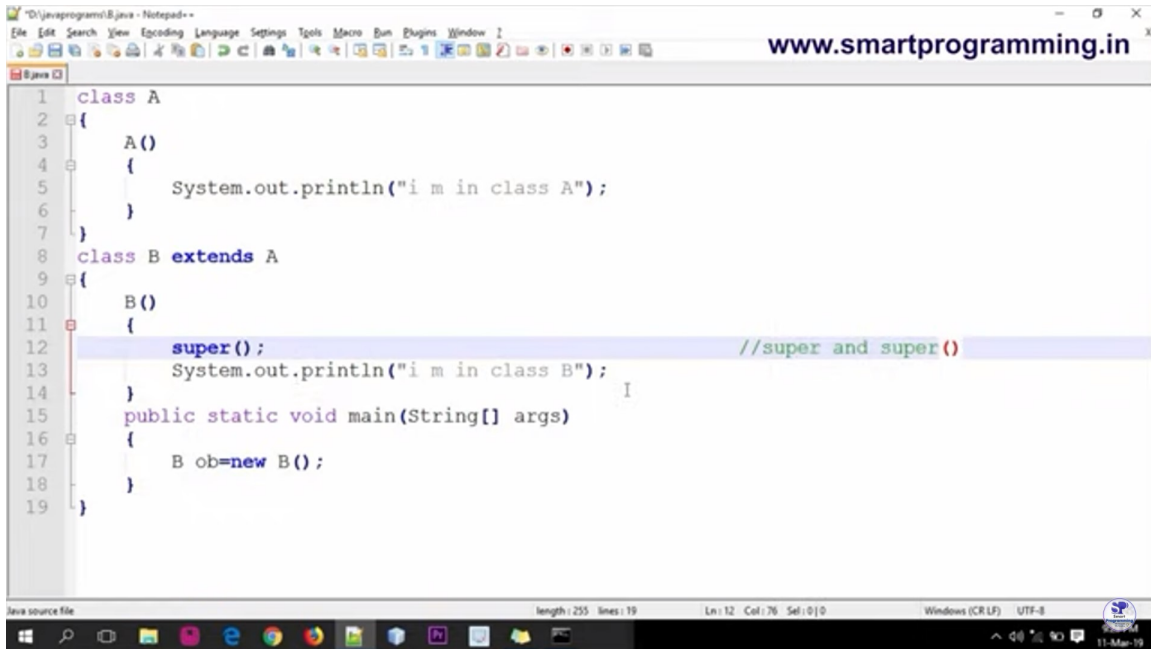
```
1 class A
2 {
3     void m1()
4     {
5         System.out.println("i m in class A");
6     }
7 }
8 class B extends A
9 {
10    void show()
11    {
12        super.m1();
13    }
14    public static void main(String[] args)
15    {
16        B ob=new B();
17        ob.show();
18    }
19 }
```

Java source file length: 216 lines: 19 Ln: 10 Col: 16 Sel: 0|0 Windows (CR LF) UTF-8 11-Mar-19

```
1 class A
2 {
3     void m1()
4     {
5         System.out.println("i m in class A");
6     }
7 }
8 class B extends A
9 {
10    void m1()
11    {
12        System.out.println(i m in class B);
13    }
14    void show()
15    {
16        m1();
17        super.m1();
18    }
19    public static void main(String[] args)
20    {
21        B ob=new B();
22        ob.show();
23    }
24 }
```

Constructors in Java | Use of Constructors | Types of Constructor...  
Java source file length: 284 lines: 24 Ln: 3 Col: 12 Sel: 2|11 Windows (CR LF) UTF-8 11-Mar-19

## Super() USED TO CALL IMMEDIATE PARENT CLASS CONSTRUCTOR



```
1 class A
2 {
3     A()
4     {
5         System.out.println("i m in class A");
6     }
7 }
8 class B extends A
9 {
10     B()
11     {
12         super(); //super and super()
13         System.out.println("i m in class B");
14     }
15     public static void main(String[] args)
16     {
17         B ob=new B();
18     }
19 }
```

The screenshot shows a Java IDE with two panels. The left panel displays the source code for two classes, A and B. Class A has a constructor A() that prints "i m in class A". Class B extends A and has its own constructor B() that prints "i m in class B". A main method in class B creates an instance of B. The right panel shows the command prompt output, which includes compilation and execution commands, followed by the runtime output of the program.

```
1 class A
2 {
3     A()
4     {
5         System.out.println("i m in class A");
6     }
7 }
8 class B extends A
9 {
10    B()
11    {
12        System.out.println("i m in class B");
13    }
14    public static void main(String[] args)
15    {
16        B ob=new B();
17    }
18 }
```

```
}
^
12 errors
D:\javaprograms>javac B.java
D:\javaprograms>java B
i m in class B
i m in class A
D:\javaprograms>javac B.java
D:\javaprograms>java B
i m in class A
i m in class B
D:\javaprograms>
```

