

## => **super keyword :**

-> The super keyword is a reference variable which is used to refer immediate parent class object

### **-> Use of super keyword :-**

**1. super keyword can be used to refer the immediate parent class instance variable**

```
class A
```

```
{  
    int no=10;  
}
```

```
class B extends A
```

```
{  
    int no=20;  
    void show(int no)  
    {  
        System.out.println(no);  
        System.out.println(this.no);  
        System.out.println(super.no);  
    }  
}
```

```
class SuperTest
```

```
{  
    public static void main(String[] args)  
    {  
        B ob=new B();  
        ob.show(30);  
    }  
}
```

## 2. super keyword can be used to invoke parent class method

```
class A
{
    void show1()
    {
        System.out.println("i m in class A");
    }
}
class B extends A
{
    void show2()
    {
        super.show1();
        System.out.println("im in class B");
    }
}
```

```
class SuperTest2
{
    public static void main(String[] args)
    {
        B ob=new B();
        ob.show2();
    }
}
```

=====

```
class A
{
    void show()
    {
        System.out.println("i m in class A");
    }
}
```

```
class B extends A
{
    void show()
    {
        System.out.println("im in class B");
    }
    void m1()
    {
        //show() ; // im in class B
        System.out.println("1");
        super.show(); //i m in class A
        System.out.println("2");
    }
}
```

```
class SuperTest3
{
    public static void main(String[] args)
    {
        B ob=new B();
        ob.m1();
    }
}
```

### 3. super keyword is used to invoke parent class constructor

```
class A
{
    A()
    {
        System.out.println("i am A class constructor");
    }
}
class B extends A
{
    B()
    {
        //super() ; --> compiler provides it implicitly
        System.out.println("i am in B class constructor");
    }
}

class SuperTest4
{
    public static void main(String[] args)
    {
        B ob=new B();
    }
}
```

- > If we dont provide super() in constructor then compiler provides it implicitly
- > super() should always be used to be the first statement in constructor but after JDK 22+, this() and super() can be used after other statements also

Points to remember :-

- > We cannot use this() and super() together

```
class A
{
    A()
    {
        System.out.println("i am A class constructor");
    }
}
```

```
class B extends A
{
    B()
    {
        System.out.println("i am in B class constructor");
        //super(); error because call to super must be first statement in constructor
    }
}
```

```
class SuperTest5
{
    public static void main(String[] args)
    {
        B ob=new B();
    }
}
```

-> this, super used for variable and method, whereas this(), super() used for constructor call

## What is difference between this and this()

	this	this()
1.	“this” is a keyword which refers to the current class object.	“this()” is this keyword along with the parenthesis used to invoke current class constructor.
2.	It is used to access instance variable of current class.	It is used to invoke current class constructor.
3.	It is used in method chaining.	It is used in constructor chaining.

## What is difference between super and super()

	super	super()
1.	“super” is a keyword mainly used to access properties and behavior of immediate parent class.	It is “super” keyword with parenthesis used to invoke the constructor of immediate parent class.