

this keyword

```
1 public void show(int a, int b){ // formal parameters
```

```
2  
3 }
```

```
4
```

```
5 show(a,b); //actuals parameters|
```


- The **this** keyword can be used to refer to any member of the current object from within an instance method or a constructor.
- '**this**' is an internal or implicit object created by JAVA for two purposes. They are
 - **this** object is internally pointing to current class object.
 - Whenever the formal parameters and data members of the class are similar, to differentiate the data members of the class from formal parameters, the data members of class must be preceded by '**this**'.

We can use this keyword in following ways

this variable

this constructor

this method

this variable

- **this** keyword can be very useful in the handling of Variable Hiding.
- We can not create two instance or local variables with the same name.

Ex:


```
1 class Test
2 {
3     int a;
4     int b;
5     Test(int a, int b) {
6         this.a = a;
7         this.b = b;
8     }
9     void display() {
10        //Displaying value of variables a and b
11        System.out.println("a = " + a + " b = " + b);
12    }
13    public static void main(String[] args) {
14        Test object = new Test(40, 80);
15        object.display();
16    }
17 }
```

this constructor

this keyword can be used inside the constructor to call another overloaded constructor in the same class.

Ex:


```
1 class Data
2 {
3     int a;
4     int b;
5     Data()
6     {
7         this(10,20);
8     }
9     Data(int x,int y)
10    {
11        a=x;
12        b=y;
13    }
14    void show()
15    {
16        System.out.println("a="+a);
17        System.out.println("b="+b);
18    }
19 }
20
21 class Test {
22     public static void main(String[] args) {
23         Data d1=new Data();
24         Data d2=new Data(100,200);
```



```
1 class Data
2 {
3     int a;
4     int b;
5     Data() {
6         this(10,20);
7     }
8     Data(int x,int y) {
9         a=x;
10        b=y;
11    }
12    void show()
13    {
14        System.out.println("a="+a);
15        System.out.println("b="+b);
16    }
17 }
18 class Test {
19     public static void main(String[] args) {
20         Data d1=new Data();
21         Data d2=new Data(100,200);
22         d1.show();
23         d2.show();
24     }
```


this method

this keyword can also be used inside Methods to call another method from same class.

Ex:

```
1 class Data
2 {
3     int a;
4     int b;
5     Data() {
6         this(10,20);
7     }
8     Data(int x,int y) {
9         a=x;
10        b=y;
11        this.show();
12    }
13    void show()
14    {
15        System.out.println("a="+a);
16        System.out.println("b="+b);
17    }
18 }
19 class Test {
20     public static void main(String[] args) {
21         Data d1=new Data();
22         Data d2=new Data(100,200);
23     }
```