# ‘this’ reference in Java

Difficulty Level : Easy  Last Updated : 27 Jul, 2021

‘this’ is a reference variable that refers to the current object.

Following are the ways to use ‘this’ keyword in java :

**1. Using ‘this’ keyword to refer current class instance variables**

```
//Java code for using 'this' keyword to
//refer current class instance variables
class Test
{
    int a;
    int b;

    // Parameterized constructor
    Test(int a, int b)
    {
        this.a = a;
        this.b = b;
    }

    void display()
    {
        //Displaying value of variables a and b
        System.out.println("a = " + a + "  b = " + b);
    }

    public static void main(String[] args)
    {
        Test object = new Test(10, 20);
        object.display();
    }
}
```

Output:

```
a = 10  b = 20
```

**2. Using this() to invoke current class constructor**

```
// Java code for using this() to
// invoke current class constructor
class Test
{
    int a;
    int b;

    //Default constructor
    Test()
    {
        this(10, 20);
        System.out.println("Inside  default constructor \n");
    }

    //Parameterized constructor
    Test(int a, int b)
    {
        this.a = a;
        this.b = b;
        System.out.println("Inside parameterized constructor");
    }

    public static void main(String[] args)
    {
        Test object = new Test();
    }
}
```

Output:

```
Inside parameterized constructor
Inside  default constructor
```

### 3. Using ‘this’ keyword to return the current class instance

```
//Java code for using 'this' keyword
//to return the current class instance
class Test
```

```
{
    int a;
    int b;

    //Default constructor
    Test()
    {
        a = 10;
        b = 20;
    }

    //Method that returns current class instance
    Test get()
    {
        return this;
    }

    //Displaying value of variables a and b
    void display()
    {
        System.out.println("a = " + a + "  b = " + b);
    }

    public static void main(String[] args)
    {
        Test object = new Test();
        object.get().display();
    }
}
```

Output:

```
a = 10  b = 20
```

### 4. Using ‘this’ keyword as method parameter

```
// Java code for using 'this'
// keyword as method parameter
class Test
{
    int a;
    int b;

    // Default constructor
```

```
    Test()
    {
        a = 10;
        b = 20;
    }

    // Method that receives 'this' keyword as parameter
    void display(Test obj)
    {
        System.out.println("a = " +obj.a + "  b = " + obj.b);
    }

    // Method that returns current class instance
    void get()
    {
        display(this);
    }

    public static void main(String[] args)
    {
        Test object = new Test();
        object.get();
    }
}
```

Output:

```
a = 10  b = 20
```

### 5. Using ‘this’ keyword to invoke current class method

```
// Java code for using this to invoke current
// class method
class Test {

    void display()
    {
        // calling function show()
        this.show();

       System.out.println("Inside display function");
    }

    void show() {
```

```
        System.out.println("Inside show function");
    }


    public static void main(String args[]) {
        Test t1 = new Test();
        t1.display();
    }
}
```

Output:

```
Inside show function
Inside display function
```

### 6. Using ‘this’ keyword as an argument in the constructor call

```
// Java code for using this as an argument in constructor
// call
// Class with object of Class B as its data member
class A
{
    B obj;

    // Parameterized constructor with object of B
    // as a parameter
    A(B obj)
    {
        this.obj = obj;

     // calling display method of class B
        obj.display();
    }

}

class B
{
    int x = 5;

    // Default Constructor that create a object of A
    // with passing this as an argument in the
   // constructor
    B()
```

```
        {
            A obj = new A(this);
        }

        // method to show value of x
        void display()
        {
            System.out.println("Value of x in Class B : " + x);
        }

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

Output:

```
Value of x in Class B : 5
```

This article is contributed by **Mehak Narang** and **Amit Kumar**.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.