'This' Keyword in JAVA

'this' is a keyword in Java. It can be used inside the method or constructor of a class. It (this) works as a reference to the current object, whose method or constructor is being invoked. This keyword can be used to refer to any member of the current object from within an instance method or a constructor.

this keyword with a field (Instance Variable)

'this' keyword can be very useful in the handling of Variable Hiding. It is not possible to create two instances/local variables with the same name. However, it is legal to create one instance variable & one local variable or Method parameter with the same name. In this scenario, the local variable will hide the instance variable this is called Variable Hiding.

```java
class JBT {

    int variable = 5;

    public static void main(String args[]) {
        JBT obj = new JBT();

        obj.method(20);
        obj.method();
    }

    void method(int variable) {
        variable = 10;
        System.out.println("Value of Instance variable :" + this.variable);
        System.out.println("Value of Local variable :" + variable);
    }

    void method() {
        int variable = 40;
        System.out.println("Value of Instance variable :" + this.variable);
        System.out.println("Value of Local variable :" + variable);
    }
}
```

Output of the code

```
Value of Instance variable :5
Value of Local variable :10
Value of Instance variable :5
Value of Local variable :40
```

'this' Keyword with Method

'this' keyword can also be used inside Methods to call another Method from same Class.

```java
class JBT {

    public static void main(String[] args) {
        JBT obj = new JBT();
        obj.methodTwo();
    }
    void methodOne(){
        System.out.println("Inside Method ONE");
        }

    void methodTwo(){
        System.out.println("Inside Method TWO");
        this.methodOne();// same as calling methodOne()
    }
}
```

```
Inside Method TWO
Inside Method ONE
```

In Summary, 'this' keyword can be used for the following:

1. this can be used to refer current class instance variable.
2. this can be used to invoke current class method (implicitly)
3. this() can be used to invoke current class constructor.
4. this can be passed as an argument in the method call.
5. this can be passed as argument in the constructor call.
6. this can be used to return the current class instance from the method.