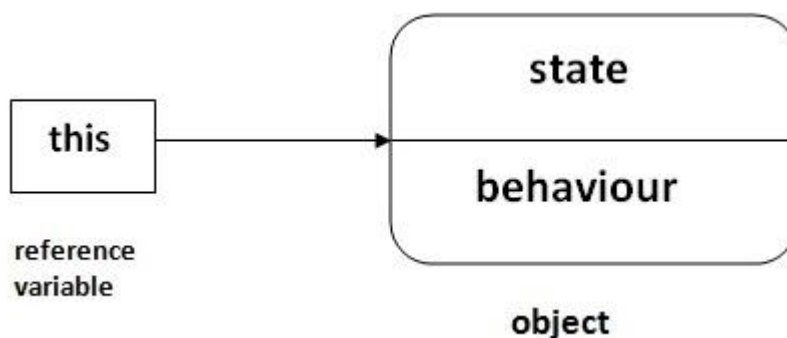


Usage of java this keyword

Here is given the 6 usage of java this keyword.

1. this keyword can be used to refer current class instance variable.
2. this() can be used to invoke current class constructor.
3. this keyword can be used to invoke current class method (implicitly)
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 keyword can also be used to return the current class instance.

Suggestion: If you are beginner to java, lookup only two usage of this keyword.



1) The this keyword can be used to refer current class instance variable.

If there is ambiguity between the instance variable and parameter, this keyword resolves the problem of ambiguity.

Understanding the problem without this keyword

Let's understand the problem if we don't use this keyword by the example given below:

- 1.
- 2.

```

3. class Student10{
4.     int id;
5.     String name;
6.
7.     Student10(int id,String name)
8. {
9.     id = id;
10.    name = name;
11.    }
12.    void display()
13.    {
14.    System.out.println(id+" "+name);
15.    }
16.
17.    public static void main(String args[]){
18.    Student10 s1 = new Student10(111,"Karan");
19.    Student10 s2 = new Student10(321,"Aryan");
20.    s1.display();
21.    s2.display();
22.    }
23. }

```

Test it Now

```

Output:0 null
       0 null

```

In the above example, parameter (formal arguments) and instance variables are same that is why we are using this keyword to distinguish between local variable and instance variable.

Solution of the above problem by this keyword

```

1. //example of this keyword
2. class Student11{
3.     int id;
4.     String name;

```

```

5.
6.   Student11(int id,String name)
7. {
8.     this.id = id;
9.     this.name = name;
10. }
11.     void display()
12. {
13.     System.out.println(id+" "+name);
14. }
15.     public static void main(String args[]){
16.         Student11 s1 = new Student11(111,"Karan");
17.         Student11 s2 = new Student11(222,"Aryan");
18.         s1.display();
19.         s2.display();
20.     }
21. }

```

2) this() can be used to invoked current class constructor.

The this() constructor call can be used to invoke the current class constructor (constructor chaining). This approach is better if you have many constructors in the class and want to reuse that constructor.

```

1. //Program of this() constructor call (constructor chainin
   g)
2.
3. class Student13{
4.     int id;

```

```

5. String name;
6. Student13()
7. {
8. System.out.println("default constructor is invoked");
9. }
10.
11. Student13(int id,String name)
12. {
13.     this();//it is used to invoked current class const
    ructor.
14.     this.id = id;
15.     this.name = name;
16. }
17. void display()
18. {
19. System.out.println(id+" "+name);
20. }
21.
22. public static void main(String args[])
23. {
24.     Student13 e1 = new Student13(111,"karan");
25.     Student13 e2 = new Student13(222,"Aryan");
26.     e1.display();
27.     e2.display();
28. }
29. }

```

3)The this keyword can be used to invoke current class method (implicitly).

```

1. class S{
2. void m()
3. {
4. System.out.println("method is invoked");
5. }

```

```
6. void n(){
7. this.m();//no need because compiler does it for you.
8. }
9. void p(){
10.    n();//complier will add this to invoke n() method
    as this.n()
11. }
12. public static void main(String args[]){
13.    S s1 = new S();
14.    s1.p();
15. }
16. }
```