



SNS COLLEGE OF ENGINEERING



Kurumbapalayam(Po), Coimbatore - 641 107

Accredited by NAAC-UGC with 'A' Grade

Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

19CS204 OBJECT ORIENTED PROGRAMMING

I YEAR /II SEMESTER

Unit 2- BASIC FEATURES OF JAVA

this- Keyword





this



this

- this is a reference variable that refers to the current object.
- this is a keyword which represents an object in a method or a constructor.
- It is basically used to eliminate the confusion between class attributes and parameters with the same name.



this



- The main motto of using this keyword is to differentiate the formal parameter and data members of the class.
- If in case, the formal parameter and data members of the class are the same, then it leads to ambiguity.
- So, in order to differentiate between formal parameter and data member of the class, the data member of the class must be preceded by the “this” keyword.

Basically, “this” keyword can be used in two ways.

1. this.
2. this()



this

1. this.

- It can be used to differentiate variable of the class and formal parameters of method or constructor. Not only that, it always points to the current class object.

Syntax

`this.data` member of the current class

2. this()

- It can be used to call one constructor within another without creating the objects multiple times for the same class.

Syntax

`this();` // call no parametrized or default constructor

`this(value1,value2,.....)` //call parametrized constructor



this



Usage of java this keyword

Here is given the 6 usage of java this keyword.

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.



this



1.this: to refer current class instance variable

- The this keyword can be used to refer current class instance variable.
- If there is ambiguity between the instance variables and parameters, this keyword resolves the problem of ambiguity.
- this keyword can be very helpful in Variable Hiding.
- Here, you cannot create two instances/local variables with the same name.
- However, it is possible to create one instance variable and one local variable with the same name.
- In this case, the local variable will be able to hide the instance variable. This is called Variable Hiding.

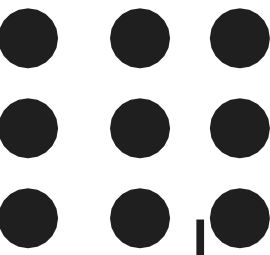


this



1.this: to refer current class instance variable

- The this keyword can be used to refer current class instance variable.
- If there is ambiguity between the instance variables and parameters, this keyword resolves the problem of ambiguity.
- this keyword can be very helpful in Variable Hiding.
- Here, you cannot create two instances/local variables with the same name.
- However, it is possible to create one instance variable and one local variable with the same name.
- In this case, the local variable will be able to hide the instance variable. This is called Variable Hiding.



this

Example

```
class Student
{
int rollno;
String name;
float fee;
Student(int rollno,String name,float fee)
{
this.rollno=rollno;
this.name=name;
this.fee=fee;
}
void display()
{
System.out.println(rollno+" "+name+" "+fee);
}
}
```

```
class TestThis2
{
public static void main(String args[])
{
Student s1=new Student(111,"ankit",500);
Student s2=new Student(112,"sumit",6000);
s1.display();
s2.display();
}
}
```




this



2) **this**: to invoke current class method

- You may invoke the method of the current class by using the **this** keyword.
- If you don't use the **this** keyword, compiler automatically adds **this** keyword while invoking the method.



this



```
Example
class Test{
void meth()
{
System.out.println("hello meth");
}
void replica(){
System.out.println("hello replica");
//meth();
this.meth();
}
}
class TestThis{
public static void main(String args[]){
Test a=new Test();
a.replica();
}}
```



this



3) **this()** : to invoke current class constructor

- The `this()` constructor call can be used to invoke the current class constructor.
- It is used to reuse the constructor. In other words, it is used for constructor chaining.



this



Example

Calling default constructor from parameterized constructor:

```
class Test{
Test()
{
System.out.println("hello default constructor");
}
Test(int x)
{
this();
System.out.println(x);
}
}
class TestThis{
public static void main(String args[]){
Test a=new Test(10);
}}
```

this keyword/ IT /SNSCE



this



Calling parameterized constructor from default constructor:

```
class Test{
Test()
{
this(5);
System.out.println("hello default constructor");
}
Test (int x)
{
System.out.println(x);
}
}
class TestThis{
public static void main(String args[]){
Test a=new Test();
}}
```



THANK YOU

this keyword/ IT /SNSCE

14/14