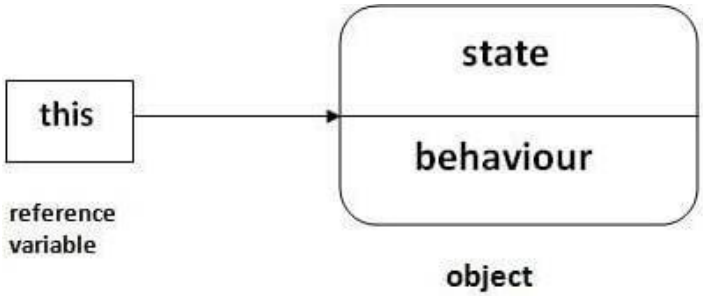
UNIT III
OBJECTANDCLASSES

thiskeywordinJava

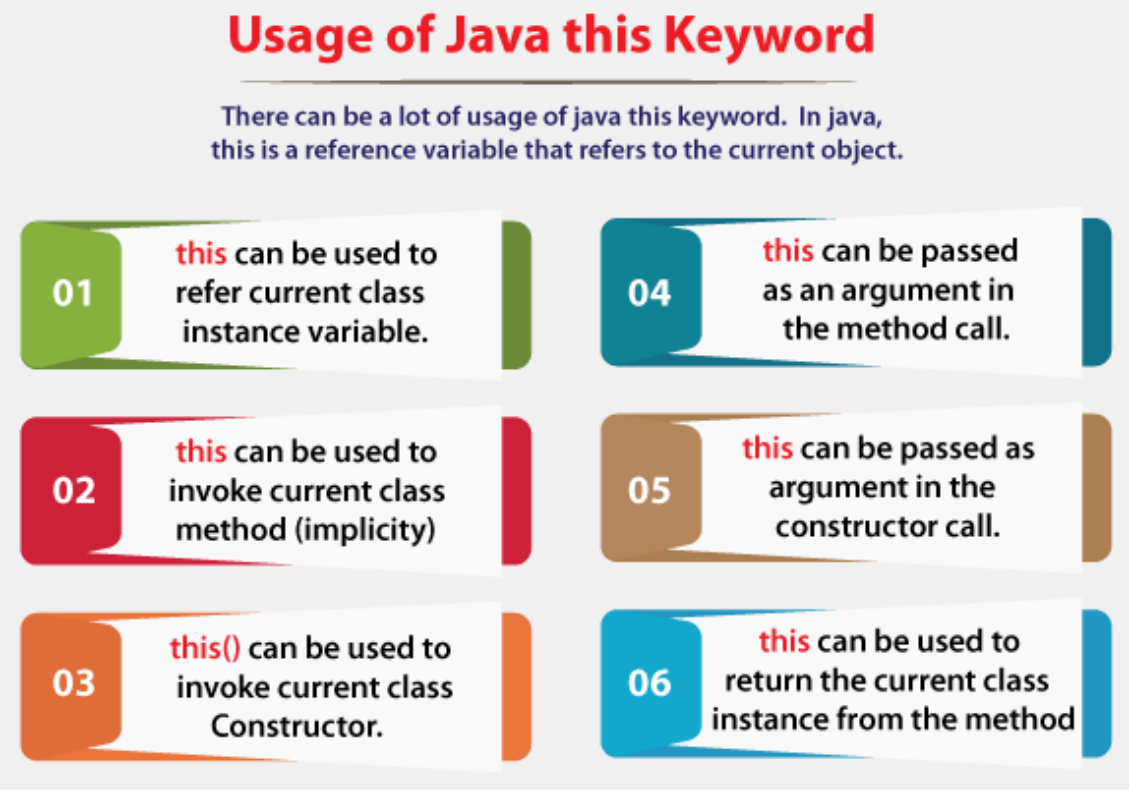
Therecanbealotofusageof**Javathiskeyword**.InJava,thisisa **reference variable** that refers to the current object.



UsageofJavathis keyword

Here isgiventhe6usageofjavathiskeyword.

1. thiscanbeusedtorefercurrentclassinstancevariable.
2. thiscanbeusedtoinvokecurrentclass method(implicitly)
3. this()canbeusedtoinvokecurrentclassconstructor.
4. thiscanbepassedasanargument inthemethodcall.
5. thiscanbepassedasargumentintheconstructorcall.
6. thiscanbeusedtoreturnthecurrentclassinstancefromthe method.



1) this:torefercurrentclassinstancevariable

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.

Understandingtheproblemwithoutthiskeyword

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

```
classStudent{
int rollno;
String name;
float fee;
Student(introllno,Stringname,floatfee){
rollno=rollno;
name=name;
fee=fee;
}
voiddisplay(){System.out.println(rollno+""+name+""+fee);}
}
classTestThis1{
public static void main(String args[]){
Students1=newStudent(111,"ankit",5000f);
```

```
Students2=newStudent(112,"sumit",6000f); s1.display();
s2.display();
}}
```

Output:

```
0null0.0
0null0.0
```

Solutionoftheaboveproblembythiskeyword class

```
Student{
int rollno;
Stringname;
float fee;
Student(introllno,Stringname,floatfee){
this.rollno=rollno;
this.name=name;
this.fee=fee;
}
voiddisplay(){System.out.println(rollno+""+name+""+fee);}
}

classTestThis2{
public static void main(String args[]){
Student s1=new Student(111,"ankit",5000f);
Students2=newStudent(112,"sumit",6000f);
s1.display();
s2.display();
}}
```

Output:

```
111ankit5000.0
112sumit6000.0
```

If local variables(formal arguments) and instance variables are different, there is no need to use this keyword like in the following program:

Programwherethiskeywordisnotrequired

```
classStudent{
int rollno;
String name;
float fee;
Student(intr,Stringn,floatf){
rollno=r;
name=n;
fee=f;
}
voiddisplay(){System.out.println(rollno+""+name+""+fee);}
}

classTestThis3{
public static void main(String args[]){
Student s1=new Student(111,"ankit",5000f);
Students2=newStudent(112,"sumit",6000f);
s1.display();
s2.display();
}}
```

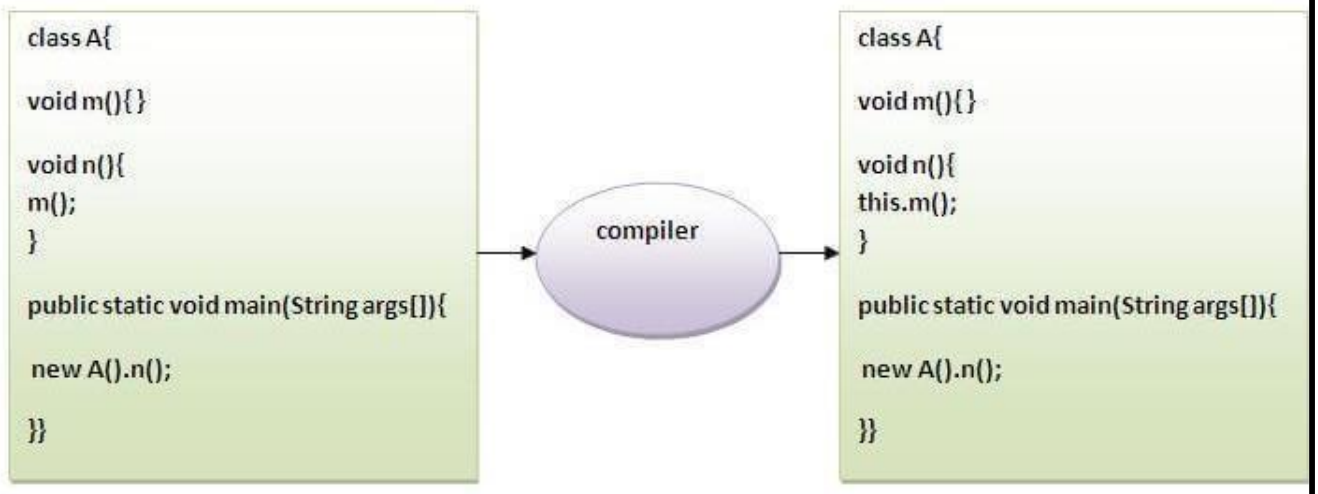
Output:

```
111ankit5000.0
112sumit6000.0
```

Itisbetterapproachtousemeaningfulnamesforvariables.Soweusesamenamefor instance variables and parameters in real time, and always use this keyword.

2) this:toinvokecurrentclassmethod

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. Let's see the example




```
class A{
void m(){System.out.println("hellom");}
void n(){
System.out.println("hellon");
//m();//sameasthis.m()
this.m();
}
}
class TestThis4{
publicstaticvoid main(Stringargs[]){ A
a=new A();
a.n();
}}
```

Output:

hello n
hellom

3) this():toinvokecurrentclassconstructor

Thethis()constructorcallcanbeusedtoinvokethecurrentclassconstructor.Itis used to reuse the constructor. In other words, it is used for constructor chaining.

Callingdefaultconstructorfromparameterizedconstructor: class

```
A{
A(){System.out.println("helloa");}
```

```
A(intx){
this();
System.out.println(x);
}
}
classTestThis5{
publicstaticvoidmain(Stringargs[]){ A
a=new A(10);
}}
```

Output:

helloa
10

Callingparameterizedconstructorfromdefaultconstructor:

```
classA{
A(){
this(5);
System.out.println("helloa");
}
A(intx){
System.out.println(x);
}
}
classTestThis6{
publicstaticvoidmain(Stringargs[]){
Aa=newA();
}}
```

Output:

5
helloa

Realusageofthis()constructorcall

The this() constructor call should be used to reuse the constructor from the constructor. It maintains the chain between the constructors i.e. it is used for constructor chaining. Let's see the example given below that displays the actualuse of this keyword.

```
classStudent{
introllno;
Stringname,course;
floatfee;
Student(introllno,Stringname,Stringcourse){
this.rollno=rollno;
this.name=name;
this.course=course;
}
Student(introllno,Stringname,Stringcourse,floatfee){
this(rollno,name,course);//reusing constructor
this.fee=fee;
}
voiddisplay(){System.out.println(rollno+""+name+""+course+""+fee);}
}
classTestThis7{
public static void main(String args[]){
Students1=newStudent(111,"ankit","java");
Students2=newStudent(112,"sumit","java",6000f);
s1.display();
s2.display();
}}
```

Output:

```
111 ankitjava0.0
112 sumitjava6000.0
```

Rule:Calltothis()mustbethe firststatement inconstructor.

```
classStudent{
introllno;
Stringname,course;
floatfee;
Student(introllno,Stringname,Stringcourse){
this.rollno=rollno;
```

```
this.name=name;
this.course=course;
}
Student(introllno,Stringname,Stringcourse,floatfee){
this.fee=fee;
this(rollno,name,course);//C.T.Error
}
voiddisplay(){System.out.println(rollno+""+name+""+course+""+fee);}
}
classTestThis8{
public static void main(String args[]){
Students1=newStudent(111,"ankit","java");
Students2=newStudent(112,"sumit","java",6000f);
s1.display();
s2.display();
}}
```

Output:

CompileTimeError:Calltothismustbefirststatementinconstructor

---

4) this:topassasanargumentinthe method

The this keyword can also be passed as anargument in the method. It is mainly used in the event handling. Let's see the example:

```
classS2{
  void m(S2 obj){
  System.out.println("methodisinvoked");
  }
  voidp(){
  m(this);
  }
  publicstaticvoidmain(Stringargs[]){
  S2 s1 = new S2();
  s1.p();
  }
}
```

Output:

methodisinvoked

Applicationofthisthatcanbepassedasanargument:

In event handling (or) in a situation where we have to provide reference of a class to another one. It is used to reuse one object in many methods.

---

5) this:topassasargumentintheconstructor call

Wecan passthethiskeywordin theconstructoralso.Itisusefulifwehavetouse one object in multiple classes. Let's see the example:

```
class B{
  A4obj;
  B(A4obj){
    this.obj=obj;
  }
  voiddisplay(){
    System.out.println(obj.data);//usingdatamemberofA4class
  }
}

classA4{
  intdata=10;
  A4(){
   Bb=newB(this); b.display();
  }
  publicstaticvoidmain(Stringargs[]){
   A4 a=new A4();
  }
}
```

Output:10

---

6) this keywordcanbeusedtoreturncurrentclassinstance

Wecanreturnthiskeywordasanstatementfromthemethod.Insuchcase,return type of the method must be the class type (non-primitive). Let's see the example:

Syntaxofthisthatcanbereturnedasastatement return_type

```
method_name(){
returnthis;
}
```

Exampleofthiskeywordthat youreturnasastatement fromthe method

```
class A{
AgetA(){
returnthis;
}
voidmsg(){System.out.println("Hellojava");}
}
classTest1{
publicstaticvoidmain(Stringargs[]){
newA().getA().msg();
}
}
```

Output:

Hellojava

Provingthiskeyword

Let's prove that this keyword refers to the current class instance variable. In this program, we are printing the reference variable and this, output of both variables are same.

```
classA5{
voidm(){
System.out.println(this);//printssamereferenceID
}
publicstaticvoidmain(Stringargs[]){ A5
obj=new A5();
System.out.println(obj);//printsthereferenceID
```

```
obj.m();
}
}
```

Output:

```
A5@22b3ea59

A5@22b3ea59
```