‘this’ reference in Java

In Java, ‘this’ is a reference variable that refers to the current object, or can be said “this” in Java is a keyword that refers to the current object instance. It can be used to call current class methods and fields, to pass an instance of the current class as a parameter, and to differentiate between the local and instance variables. Using “this” reference can improve code readability and reduce naming conflicts.

Java this reference Example

In Java, this is a reference variable that refers to the current object on which the method or constructor is being invoked. It can be used to access instance variables and methods of the current object.

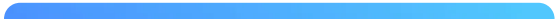
Below is the implementation of this reference:

Java

```
// Java Program to implement
```

```
public class ThisExample {
    int num = 10;

    public ThisExample() {
        System.out.println("Inside constructor");
    }

    public ThisExample(int num) {
        // Invoking default constructor
        this();

        // Assigning the local variable num to the instance variable num
        this.num = num;
    }

    void display() {
        // Invoking the method show() of the current class
        this.show();

        // Displaying the value of the instance variable num
        System.out.println("num: " + this.num);
    }

    void show() {
        System.out.println("Inside show method");
    }

    public static void main(String[] args) {
        ThisExample obj = new ThisExample(100);
        obj.display();
    }
```

Output



```
Inside constructor
Inside show method
num: 100
```

Explanation of the above Program

In the above code, we have defined a Person class with two private fields name and age. We have defined the Person class constructor to initialize these fields using this keyword. We have also defined getter and setter methods for these fields which use this keyword to refer to the current object instance.

In the printDetails() method, we have used this keyword to refer to the current object instance and print its

In the Main class, we have created two Person objects and called the printDetails() method on each object. The output shows the name, age, and object reference of each object instance.

Methods to use ‘this’ in Java

Following are the ways to use the ‘this’ keyword in Java mentioned below:

- Using the ‘this’ keyword to refer to current class instance variables.
- Using this() to invoke the current class constructor
- Using ‘this’ keyword to return the current class instance
- Using ‘this’ keyword as the method parameter
- Using ‘this’ keyword to invoke the current class method
- Using ‘this’ keyword as an argument in the constructor call

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

---

Java

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

    // Parameterized constructor
```

```
        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

```
// Java code for using this() to
// invoke current class constructor
class Test
```

```
    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
```

Java

```
//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)
    {
```

```
    }
}
```

Output:

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

4. Using ‘this’ keyword as a method parameter

Java

```
// 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)
    {
```

```
    }

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

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

Output:

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

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

Java

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

    void display()
    {
```

```
        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

```
// Java code for using this as an argument in constructor
// call
```

```
    // 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 an 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
```

Advantages

1. It helps to distinguish between instance variables and local variables with the same name.
2. It can be used to pass the current object as an argument to another method.
3. It can be used to return the current object from a method.
4. It can be used to invoke a constructor from another overloaded constructor in the same class.

Disadvantages

1. Overuse of this can make the code harder to read and understand.
2. Using this unnecessarily can add unnecessary overhead to the program.
3. Using this in a static context results in a compile-time error.
4. Overall, this keyword is a useful tool for working with objects in Java, but it should be used judiciously and only when necessary.

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.

Last Updated : 15 May, 2023

204

Similar Reads

1. java.lang.ref.Reference Class in Java
2. How to pass integer by reference in Java
3. Reference Variable in Java
4. Different Ways to Achieve Pass By Reference in Java
5. Converting ArrayList to HashMap using Method Reference in Java 8
6. Backwards Compatibility in a Software System with Systematic Reference to Java
7. Passing Strings By Reference in Java
8. Difference Between java.sql.Time, java.sql.Timestamp and java.sql.Date in Java
9. Referencing Subclass objects with Subclass vs Superclass reference
10. How to Convert java.sql.Date to java.util.Date in Java?

1. Spring MVC Tutorial
2. Spring Tutorial
3. Spring Boot Tutorial
4. Java 8 Tutorial
5. CBSE Class 11 Syllabus

Previous

Interfaces in Java

Next

Inheritance and Constructors in Java

Article Contributed By :

GeeksforGeeks

Vote for difficulty

Current difficulty : Easy

Improved By : Akanksha_Rai, imeenalgrover, arorakashish0911, simmytarika5, varshagumber28, sayanc170, anikettchpiow

Article Tags : java-basics, Java, School Programming

Practice Tags : Java

Improve Article | Report Issue

GeeksforGeeks
A-143, 9th Floor, Sovereign Corporate Tower, Sector-136, Noida, Uttar Pradesh - 201305

feedback@geeksforgeeks.org

Languages

Python

Java

C++

GoLang

SQL

R Language

Android Tutorial

Data Structures

Array

String

Linked List

Stack

Queue

Tree

Graph

Algorithms

Sorting

Searching

Greedy

Dynamic Programming

Pattern Searching

Recursion

Backtracking

Web Development

HTML

CSS

JavaScript

Bootstrap

ReactJS

AngularJS

NodeJS

Computer Science

Python

Operating Systems

Computer Network

Database Management System

Software Engineering

Digital Logic Design

Engineering Maths

Django Tutorial

Python Projects

Python Tkinter

OpenCV Python Tutorial

Python Interview Question

Data Science & ML

Data Science With Python

Data Science For Beginner

Machine Learning Tutorial

Maths For Machine Learning

Pandas Tutorial

NumPy Tutorial

NLP Tutorial

Deep Learning Tutorial

DevOps

Git

AWS

Docker

Kubernetes

Azure

GCP

Competitive Programming

Top DSA for CP

Top 50 Tree Problems

Top 50 Graph Problems

Top 50 Array Problems

System Design

What is System Design

Monolithic and Distributed SD

Scalability in SD

Databases in SD

Top 15 Websites for CP

Top SD Interview Questions

Interview Corner

Company Preparation

Preparation for SDE

Company Interview Corner

Experienced Interview

Internship Interview

Competitive Programming

Aptitude

GfG School

CBSE Notes for Class 8

CBSE Notes for Class 9

CBSE Notes for Class 10

CBSE Notes for Class 11

CBSE Notes for Class 12

English Grammar

Commerce

Accountancy

Business Studies

Microeconomics

Macroeconomics

Statistics for Economics

Indian Economic Development

UPSC

Polity Notes

Geography Notes

History Notes

Science and Technology Notes

Economics Notes

Important Topics in Ethics

UPSC Previous Year Papers

SSC/ BANKING

SSC CGL Syllabus

Write & Earn

Write an Article

IBPS PO Syllabus

IBPS Clerk Syllabus

Aptitude Questions

SSC CGL Practice Papers

Write Interview Experience

Internships

Video Internship