**Usage of java this keyword**

Here is given the 6 usage of java this keyword.

this can be used to refer current class instance variable.

this can be used to invoke current class method (implicitly)

this() can be used to invoke current class constructor.

this can be passed as an argument in the method call.

this can be passed as argument in the constructor call.

this can be used to return the current class instance from the method.

Keyword THIS is a reference variable in Java that refers to the current object.

Rule:

Call to this() must be the first statement in constructor.

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.

class Test

{

    int a;

    int b;

    // Parameterized constructor

    Test(int a, int b)

    {

        this.a = a;

        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();

    }

}

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.

class Test {

     void display()

    {

        // calling fuction show()

        this.show();

        System.out.println("Inside display function");

    }

    void show() {

        System.out.println("Inside show funcion");

    }

     public static void main(String args[]) {

        Test t1 = new Test();

        t1.display();

    }

}

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.

class Test

{

    int a;

    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();

    }

}

Real Example

class Student{

int rollno;

String name,course;

float fee;

Student(int rollno,String name,String course){

this.rollno=rollno;

this.name=name;

this.course=course;

}

Student(int rollno,String name,String course,float fee){

this(rollno,name,course);//reusing constructor

this.fee=fee;

}

void display(){System.out.println(rollno+" "+name+" "+course+" "+fee);}

}

class TestThis7{

public static void main(String args[]){

Student s1=new Student(111,"ankit","java");

Student s2=new Student(112,"sumit","java",6000f);

s1.display();

s2.display();

}}

4) this: to pass as an argument in the method

The this keyword can also be passed as an argument in the method. It is mainly used in the event handling.

class Test

{

    int a;

    int b;

    // Default constructor

    Test()

    {

        a = 10;

        b = 20;

    }

      // Method that receives 'this' keyword as parameter

    void display(Test obj)

    {

        System.out.println("a = " + a + "  b = " + b);

    }

    // Method that returns current class instance

    void get()

    {

        display(this);

    }

     public static void main(String[] args)

    {

        Test object = new Test();

        object.get();

    }

}

5) this: to pass as argument in the constructor call

We can pass the this keyword in the constructor also. It is useful if we have to use one object in multiple classes.

class A

{

    B obj;

         // 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 Contructor that create a 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();

    }

}

6) this keyword can be used to return current class instance

We can return this keyword as an statement from the method. In such case, return type of the method must be the class type (non-primitive)

Syntax

return\_type method\_name(){

return this;

}

Example

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)

    {

        Test object = new Test();

        object.get().display();

    }

}