this keyword in Java

The **this** keyword is used to represent the current instance of a class. It is used to access the instance variables and invoke current class methods and constructors. The this keyword can be passed as an argument to a method call representing the current class instance.

Uses of “this” keyword in Java

- It can be used to refer to the current class instance variables.

The **this** keyword can be used to access and modify the instance and static variables. Thus, we can change the values of the variables that are declared inside the class through the this keyword. Also, this keyword can be used to differentiate between method-specific and class variables.

```
class Demo{

  // declaring an instance variable
  int instanceVar = 5;

  // declaring an static variable
  static int staticVar = 10;

  void DemoMethod() {
    // Method-specific variables
    int instanceVar = 20;
    int staticVar = 40;

    // referring to the current class instance and static variables
    this.instanceVar = 50;
    this.staticVar = 100;

    // printing the current class instance and static variable.
    System.out.println("Value of instance variable :
 " + this.instanceVar);
    System.out.println("Value of static variable : " +
   this.staticVar);

    // printing the method specific variables.
    System.out.println("instanceVar inside method : " + instanceVar);
    System.out.println("staticVar inside method: " + staticVar);
  }
}

public class Main {

  public static void main(String[] args) {
    Demo obj = new Demo();
    obj.DemoMethod();
  }
}
```

- It can be passed as an argument in the method call representing the current object of the class.

The **this** keyword can be passed as an argument in a method representing an object of a class. At industry level it is used in event handlers or at places where we have to provide reference of one class to another class

```
class Demo{

  int value = 5;
  // print method
  void print(Demo ob) {
    System.out.println("ob.value = " + ob.value);
  }

  void invoke() {
    // print method is invoked by passing this as an argument
    print(this);
  }
}

public class Main {

  public static void main(String[] args) {
    Demo obj = new Demo();
    obj.invoke();
  }
}
```

- It can be used to return the current class instance from any of its instance methods.

The **this** keyword is used to return the current instance of a class. Methods of the class can be called directly at the time of creating an object using this

```
// illustration class
class Demo{

  Demo getDemo() {
    // returing the instance of current class
    return this;
  }

  void print() {
    System.out.println("Hello World!");
  }
}

public class Main {

  public static void main(String[] args) {
    new Demo().getDemo().print();
  }
}
```

- It can be used to access instance and static variables of the current instance.
- It can be passed as an argument in the constructor call.

The “**this**” keyword can be passed as an argument in the constructor call. By passing this as an argument in the constructor call we can exchange data from one instance to another. This makes exchange of data between multiple classes or objects a lot easier.

```
// A class
class A {

  // instance variable
  B tmp;

  // parameterized constructor
  A(B tmp) {
    this.tmp = tmp;
  }

  // print method
  void print() {
    System.out.println("The number is : " + tmp.val);
  }
}

// B class
class B {

  // instance variable
  int val = 50;

  // constructor
  B() {
    // creating instance of A
    // passing “this” as an argument in constructor call
    A obj = new A(this);
    obj.print();
  }
}

public class Main {

  public static void main(String[] args) {
    B b = new B();
  }
}
```

- It is used to initiate a constructor of the current class.

The **this** keyword is used to invoke the constructors of the current class. Suppose a class has two constructors: no-arg and parametrized constructors.

```
class Demo{

  // simple constructor
  Demo() {
    // invoking parameterized constructor
    this(10);
  }

  // parameterized constructor
  Illustration(int x) {
    System.out.println("Current class parameterized constructor invoked.");
    System.out.println("Number is : " + x);
  }
}

public class Main {

  public static void main(String[] args) {
    Demo obj = new Demo();
  }
}
```

- It can be used to invoke the current class methods.

The “**this**” keyword is used to invoke the methods of the current object or class.

```
class Demo{

  // current class method
  void DemoMethod() {
    System.out.print("My name is ");
  }
  void name() {
    // invoking current class DemoMethod method.
    this.DemoMethod();
    System.out.println("Debasish.");
  }
}

public class Main {

  public static void main(String[] args) {
    Demo obj = new Demo();
    obj.name();
  }
}
```