**THIS KEYWORD**

**This** keyword refers to the object that is currently invoking a method. It has a lot of usage but we will demonstrate two use cases:   
**1)** Refer to instance variable of the class  
**2)** Invoking the constructor of the class

**Invoking the constructor of the class**  
class Employee {

int id;

String name;

int dept\_id;

Employee() { // default constructor

dept\_id = 7; // assume dept\_id is common to all employees

}

/\* parameterized constructor

\* arguments and instance variables are same ( 'this' keyword

\* removes the ambiguity ) \*/

Employee(int id, String name) {

this(); // invoke the default constructor

this.name = name;

this.id = id;

}

void displayData() {

System.out.println("Dept. Id : " + dept\_id);

System.out.println("Employee Name : " + name);

System.out.println("Employee ID : " + id);

}

}

public class ThisKeywordDemo {

public static void main(String args[]) {

Employee emp = new Employee(4417, "Rahul");

emp.displayData();

}

}

### The this keyword can be used to refer current class instance variable

|  |
| --- |
| If there is ambiguity between the instance variable and parameter, this keyword resolves the problem of ambiguity. |

#### Understanding the problem without this keyword

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

1. **class** Student10{
2. **int** id;
3. String name;
5. Student10(**int** id,String name){
6. id = id;
7. name = name;
8. }
9. **void** display(){System.out.println(id+" "+name);}
11. **public** **static** **void** main(String args[]){
12. Student10 s1 = **new** Student10(111,"Karan");
13. Student10 s2 = **new** Student10(321,"Aryan");
14. s1.display();
15. s2.display();
16. }
17. }

|  |
| --- |
| In the above example, parameter (formal arguments) and instance variables are same that is why we are using this keyword to distinguish between local variable and instance variable. |

#### Solution of the above problem by this keyword

1. //example of this keyword
2. **class** Student11{
3. **int** id;
4. String name;
6. Student11(**int** id,String name){
7. **this**.id = id;
8. **this**.name = name;
9. }
10. **void** display(){System.out.println(id+" "+name);}
11. **public** **static** **void** main(String args[]){
12. Student11 s1 = **new** Student11(111,"Karan");
13. Student11 s2 = **new** Student11(222,"Aryan");
14. s1.display();
15. s2.display();
16. }
17. }

|  |
| --- |
| If local variables(formal arguments) and instance variables are different, there is no need to use this keyword. |