## Interview Questions on This Keyword in Java

### What is this keyword ?

this is reference variable that refers to the current object.

### Why use this keyword ?

this keyword can be used to refer current class instance variable and this keyword can also be used to return the current class instance.

### What are the uses of this keyword in constructor ?

this can be passed as argument in the constructor call.

### Can we call methods using this keyword ?

Yes we can use this keyword to call current class non static methods.

### Uses of this keyword with constructor ?

Used to invoke current class constructor.

### Difference Between this() and super() ?

Super keyword is always pointing to base class features and this keyword is always pointing to current class features.

this is a reference to the current object in which this keyword is used whereas super is a reference used to access members specific to the parent Class.

this is primarily used for accessing member variables if local variables have same name, for constructor chaining and for passing itself to some method whereas super is primarily used to initialize base class members within derived class constructor.

### What is the difference between this. (this dot) and this() (this off). ?

this. can be used to differentiate variable of class and formal parameters of method or constructor.  
this() can be used to call one constructor within the another constructor without creation of objects multiple time for the same class.

### What is difference between super(), super(..), this() and this(..) ?

super() and super(..) are used for establishing the communication between base class and derived class constructor.

this() and this(...) are used for establishing the communication between current class constructor.

**1.What is this key word in java?**

* "**this**"is a predefined instance variable to hold current object reference

**2.What are the uses of this keyword in constructor?**

1.**this**must be used to access instance variable if both instance and local variable names are same.

1. package com.instanceofjava;
3. public class ThisDemo {
4. int a, b;
6. ThisDemo(int a, int b){
8. a=a;
9. b=b;
10. }
12. public static void main(String[] args){
14. ThisDemo obj= new ThisDemo(1,2);
16. System.out.println(obj.a);
17. System.out.println(obj.b);
18. }
19. }

Output:

1. 0
2. 0

using this keyword:

1. package com.instanceofjava;
3. public class ThisDemo {
4. int a, b;
6. ThisDemo(int a, int b){
8. this.a=a;
9. this.b=b;
10. }
12. public static void main(String[] args){
14. ThisDemo obj= new ThisDemo(1,2);
16. System.out.println(obj.a);
17. System.out.println(obj.b);
18. }
19. }

Output:

1. 1
2. 2

* We can use this keyword in constructor overloading.
* To call one constructor from another we need this(); and this(); call should be first statement of the constructor.

2.Used to invoke current class constructor:

1. package com.instanceofjava;
3. public class ThisDemo {
4. int a, b;
6. ThisDemo(){
7. System.out.println("Default constructor called");
8. }
9. ThisDemo(int a, int b){
10. this();
11. this.a=a;
12. this.b=b;
13. }
15. public static void main(String[] args){
17. ThisDemo obj= new ThisDemo(1,2);
19. System.out.println(obj.a);
20. System.out.println(obj.b);
21. }
22. }

Output:

1. Default constructor called
2. 1
3. 2

**3. Can we call methods using this keyword?**

* Yes we can use this keyword to call current class non static methods .

1. package com.instanceofjava;
3. public class Test{
4. int a, b;
6. Test(int a, int b){
8. this.a=a;
9. this.b=b;
10. }
12. void show(){
14. System.out.println("Show() method called");
16. }
18. void print(){
20. this.show();
21. System.out.println(a);
22. System.out.println(b);
24. }
25. public static void main(String[] args){
27. Test obj= new Test(1,2);

30. obj.print()
31. }
32. }

Output:

1. Show() method called
2. 1
3. 2

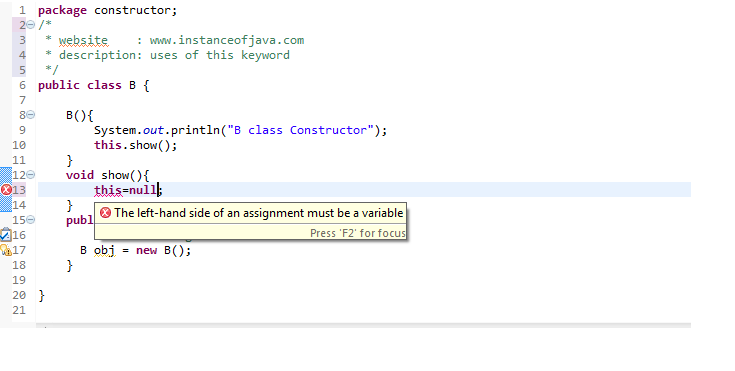
**3. Can we call method on this keyword from constructor?**

* Yes we can call non static methods from constructor using this keyword.

[](https://2.bp.blogspot.com/-y6x1Jwa6j-Y/VuAyUt9XxlI/AAAAAAAAAps/kWFngNJG0lM/s1600/this+keyword.png)

**4.Is it possible to assign reference to this ?**

* No we can not assign any value to "**this**" because its always points to current object and it is a final reference in java.
* If we try to change or assign value to **this**compile time error will come.
* The left-hand side of an assignment must be a variable

[](https://4.bp.blogspot.com/-mgjXXi8aglY/VuA0DilNpyI/AAAAAAAAAp4/h3aXok7ELRs/s1600/this+keyword2.png)

**5.Can we return this from a method?**

* Yes We can return **this**as current class object.

1. public class B{
2. int a;
4. public int getA() {
5. return a;
6. }
8. public void setA(int a) {
9. this.a = a;
10. }
12. B show(){
13. return this;
14. }
16. public static void main(String[] args) {
18. B obj = new B();
20. obj.setA(10);
22. System.out.println(obj.getA());
23. B obj2= obj.show();
24. System.out.println(obj2.getA());
26. }
27. }

Output:

1. 10
2. 10

**6.Can we pass this as parameter of method?**

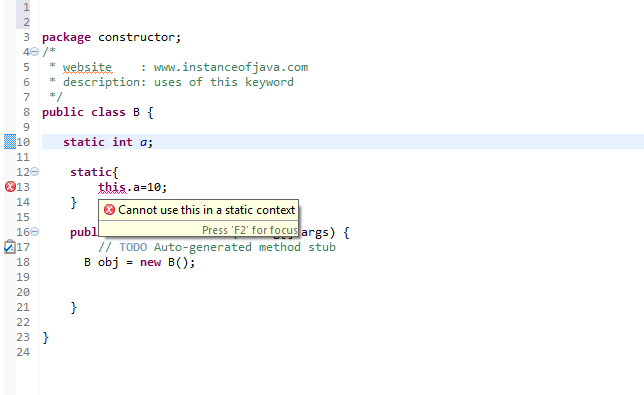
* Yes we can pass **this**as parameter in a method

**7. Can we use this to refer static members?**

* Yes its possible to access static variable of a class using this but its discouraged and as per best practices this should be used on non static reference.

**8.Is it possible to use this in static blocks?**

* No its not possible to use this keyword in static block.

[](https://1.bp.blogspot.com/-xsCmHvk-dwM/VuA4n2D9PoI/AAAAAAAAAqE/zC8M-H0O3iQ/s1600/this+keyword+in+java+static+block.png)

**9.Can we use this in static methods?**

* No we can not use this in static methods. if we try to use compile time error will come:Cannot use this in a static context

**10.What are all the differences between this and super keyword?**

* This refers to current class object where as super refers to super class object
* Using this we can access all non static methods and variables. Using super we can access super class variable and methods from sub class.
* Using this(); call we can call other constructor in same class. Using super we can call super class constructor from sub class constructor.

**11. write a java program on constructor overloading or constructor chaining  using this and super keywords**

* [Constructor chaining in java with example program](http://www.instanceofjava.com/2016/02/constructor-chaining-in-java-using-this-super.html)

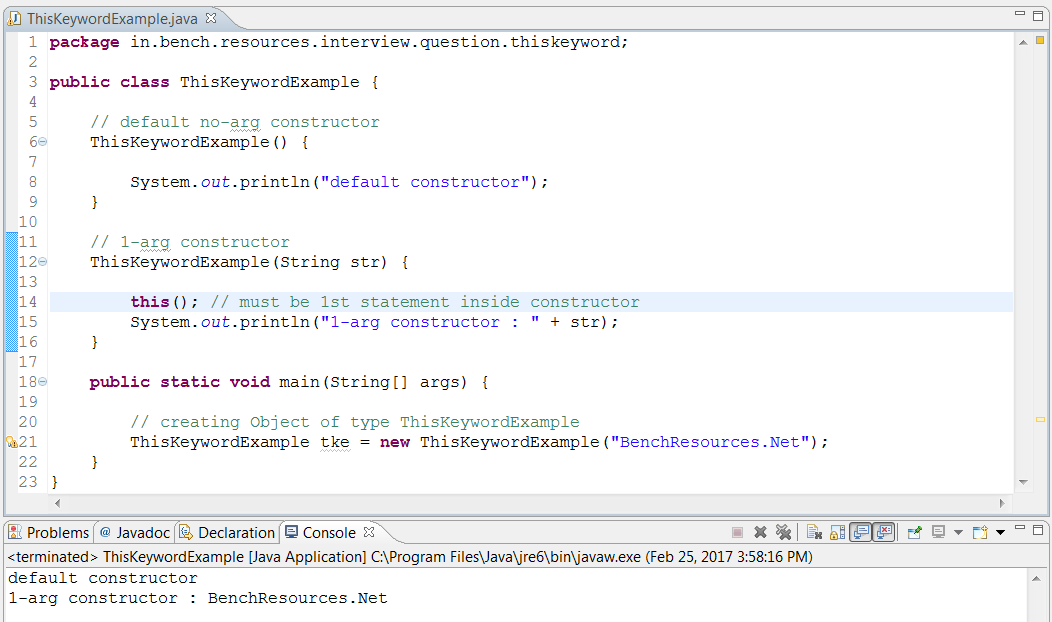
**Q) Explain the usage of this keyword ?**

**Usage of this keyword in Java:**

* **Instance variable:**this keyword is used to refer the current instance variables of the class
* **Class constructor:**this() constructor call; is used to invoke other overloaded constructor of the same class
* **Instance method:**<methodName> is used to invoke current instance method of the same class
* **Method parameter:**this keyword can be used to pass as argument in method invocation
* **Return type:**this keyword can be used to return current class instance
* **Note:**this cannot be used to refer anything in *static* context

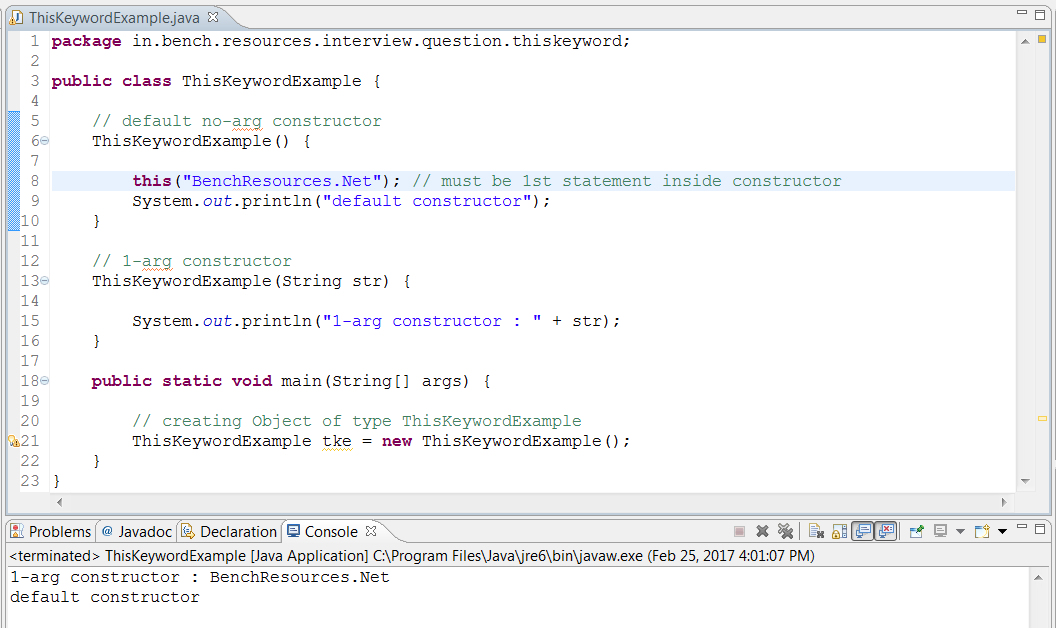
**Q) Whether it is possible to invoke one constructor from another constructor in the same class ?**

* Yes, it is possible to invoke one constructor from another constructor using this keyword
* But this() ; constructor call must be ***1st statement*** inside constructor
* Otherwise, compile-time error will be thrown stating “***Constructor call must be the first statement in a constructor***”
* Example, as shown in the below screen-capture

**[](http://www.benchresources.net/wp-content/uploads/2017/02/invoking-default-constructor-from-another-constructor-using-this-keyword-1.png)**

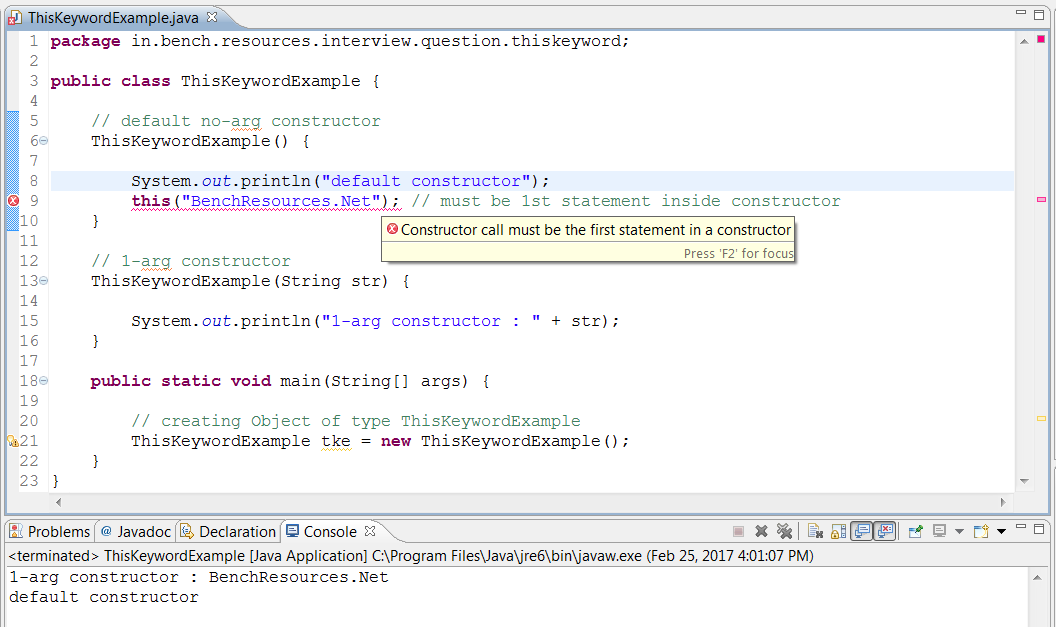
**Q) How to invoke parameterized constructor from another constructor in the same class ?**

* Using this keyword, we can also pass constructor-arguments as shown in the below screen-capture
* But this(arguments…) ; constructor call must be ***1st statement*** inside constructor
* Otherwise, ***compile-time error*** will be thrown stating “***Constructor call must be the first statement in a constructor***”
* **Syntax:** this(arguments…);

**[](http://www.benchresources.net/wp-content/uploads/2017/02/invoking-parameterized-constructor-from-another-constructor-using-this-keyword-2.png)**

**Q) What will happen, if this() constructor call is present in last line of constructor ?**

* Whenever this(); constructor call present inside constructor to invoke another constructor, then it must be ***1st statement***
* Otherwise, ***compile-time error*** will be thrown stating “***Constructor call must be the first statement in a constructor***”
* Example, as shown in the below screen-capture

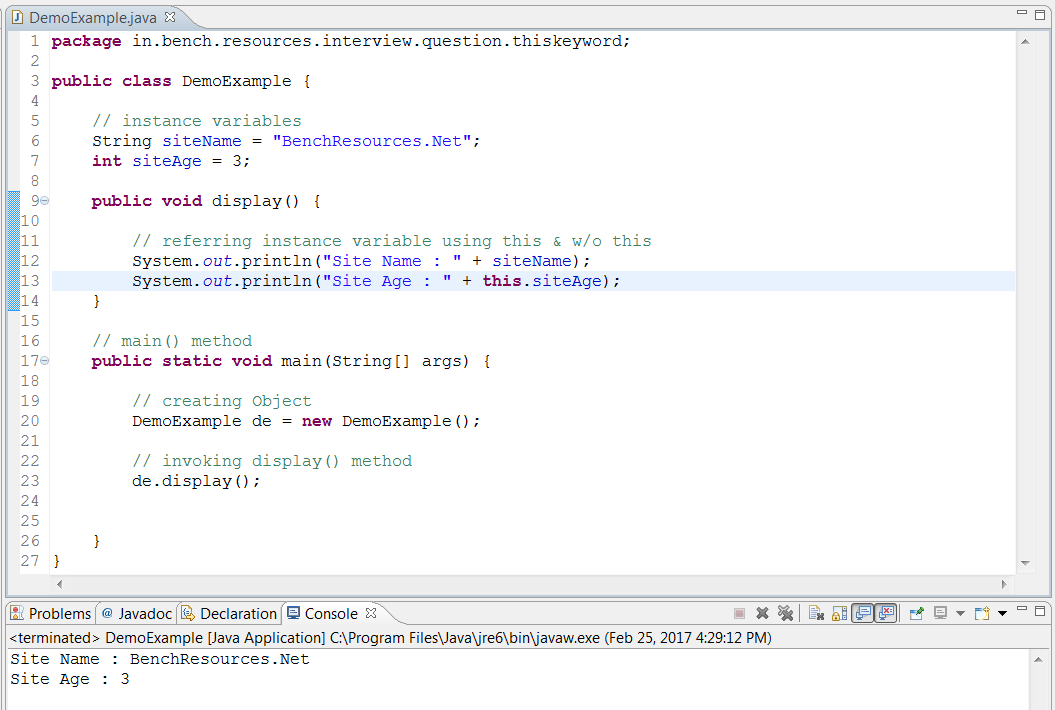
**[](http://www.benchresources.net/wp-content/uploads/2017/02/this-constructor-call-present-in-last-statement-of-constructor-3.png)**

**Q) Whether compiler inserts this(); constructor implicitly similar to super() constructor call ?**

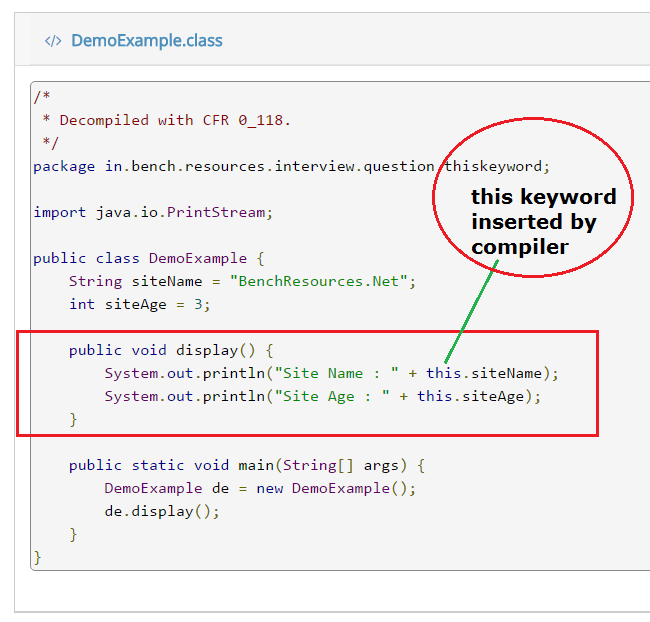
* No, compiler doesn’t inserts this(); constructor call
* Programmer can ***write***either ***super()***or ***this()*** constructor call as ***1st statement***
* Else, if it ***isn’t explicitly*** coded, then compiler inserts ***super(); constructor***call as ***1st statement*** inside constructor

**Q) How can we refer instance variables of same class ?**

* All member variables of same class can be ***referred***using ***this keyword***
* ***Directly***referring by variable name ***without***using this will ***also* *work***
* Because after ***compilation***, compiler inserts ***this keyword*** followed by ***dot notation*** and then actual ***variable name***
* Example, as shown in the below screen-capture

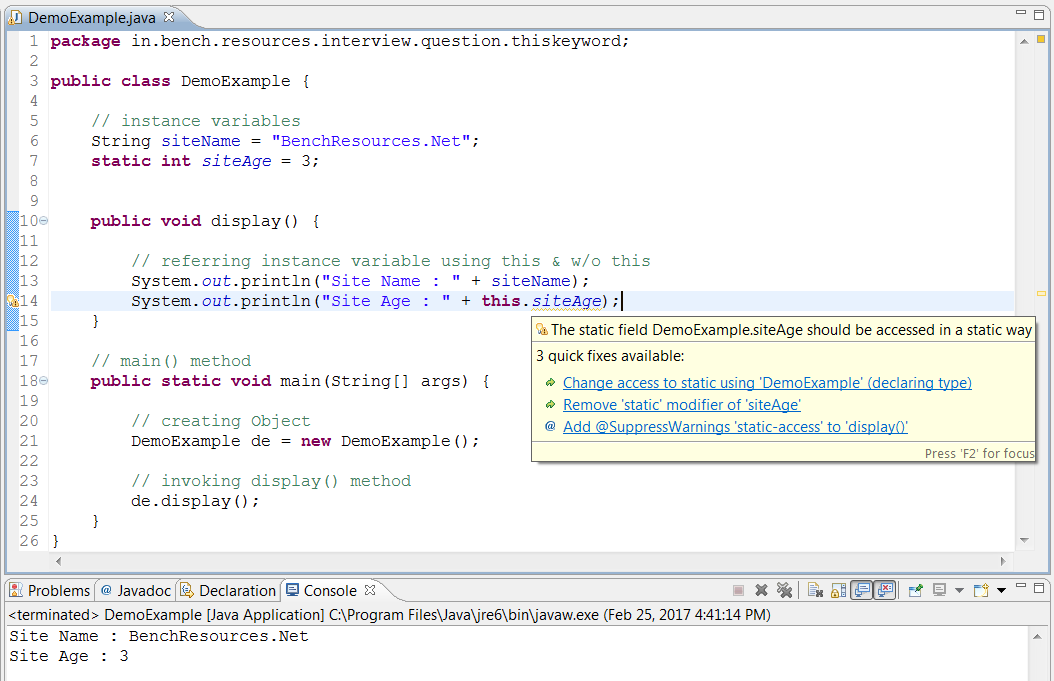
**[](http://www.benchresources.net/wp-content/uploads/2017/02/referring-variable-using-this-keyword-4.png)**

**After compilation & de-compilation:**

**[](http://www.benchresources.net/wp-content/uploads/2017/02/referring-variable-using-this-keyword-after-de-compilation-4b.png)**

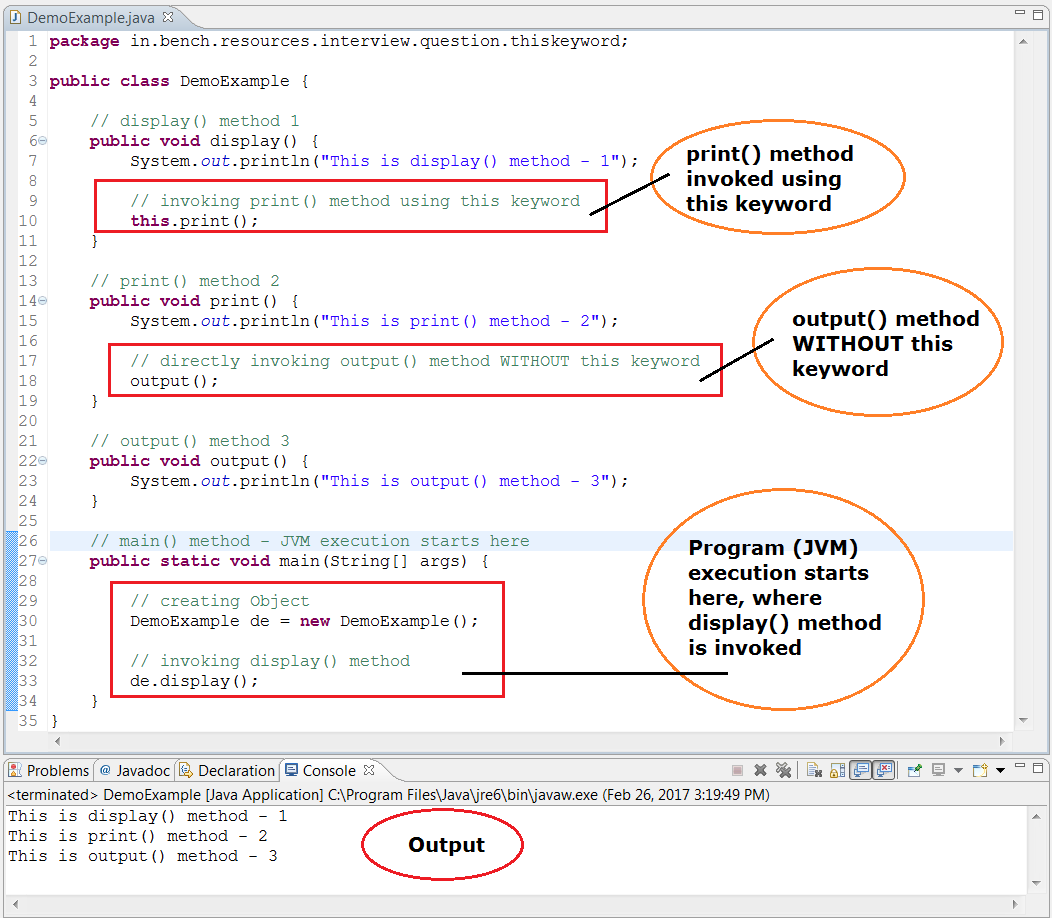
**Q) Whether it is possible to refer static variables using this keyword ?**

* Yes, it is ***possible***to refer static variable using this keyword
* But its ***usage is discouraged*** as static variable belongs to class & it need to be accessed in static way
* **Syntax:**<class-name>.<static-variable-name-of-class>
* When we try to access using this keyword, then compiler ***warns***with a message “***The static field DemoExample.siteAge should be accessed in a static way***”

**[](http://www.benchresources.net/wp-content/uploads/2017/02/referring-static-variable-using-this-keyword-5.png)**

**Q) Whether it is possible to invoke instance methods using this keyword ?**

* All instance methods of same class can be ***invoked***from another method using this keyword
* But we can also invoke instance methods ***directly***as well ***without***using ***this keyword***
* Because after ***compilation***, compiler inserts ***this keyword*** followed by ***dot notation*** and then ***instance method***
* Example, as shown in the below screen-capture

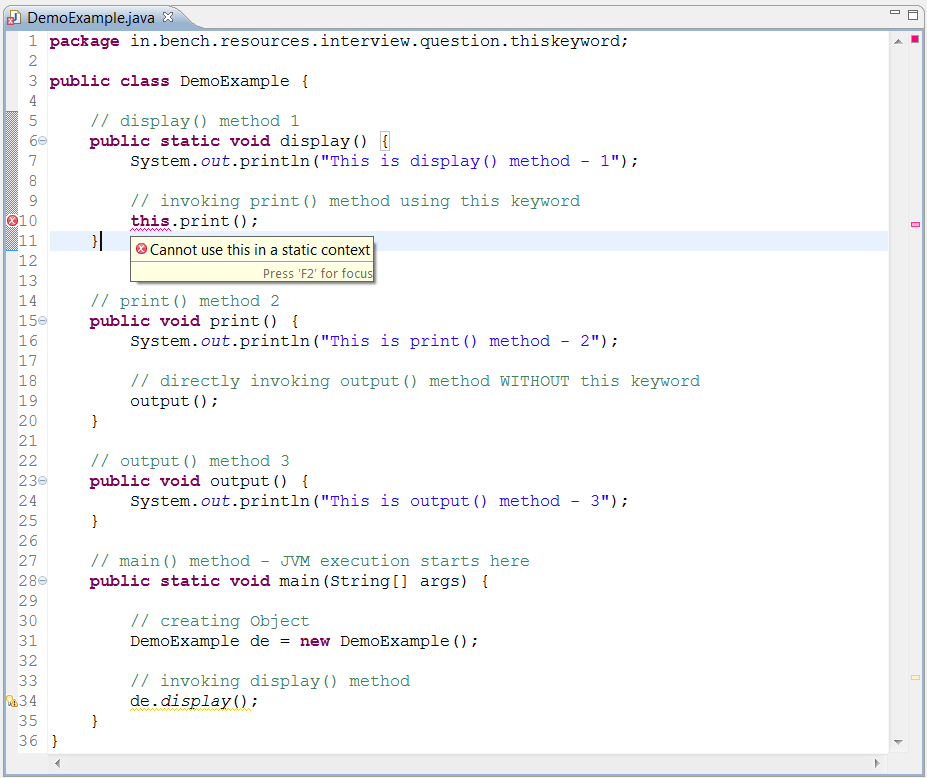
**[](http://www.benchresources.net/wp-content/uploads/2017/02/invoking-instance-method-using-this-keyword-6.png)**

**After compilation & de-compilation:**

**[](http://www.benchresources.net/wp-content/uploads/2017/02/invoking-instance-method-using-this-keyword-6b.png)**

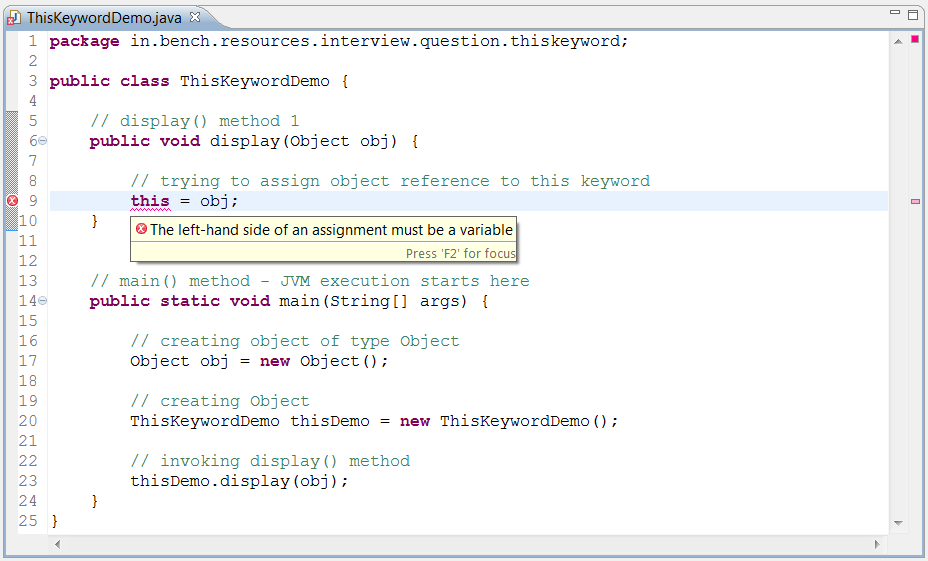
**Q) What will happen, if static methods (non-instance method) is invoked using this keyword ?**

* Whenever we try to***invoke static methods***using ***this keyword*** –> a ***compile-time error*** will be thrown stating “***Cannot use this in a static context***”
* Example, as shown in the below screen-capture

**[](http://www.benchresources.net/wp-content/uploads/2017/02/referring-static-method-using-this-keyword-7.png)**

**Q) Whether it is possible to assign references to this keyword ?**

* ***Assigning***any ***object reference*** to ***this keyword*** results in ***compile-time error*** stating “***The left-hand side of an assignment must be a variable***”
* Example, as shown in the below screen-capture

**[](http://www.benchresources.net/wp-content/uploads/2017/02/assigning-reference-to-this-keyword-throws-compile-time-error-8.png)**

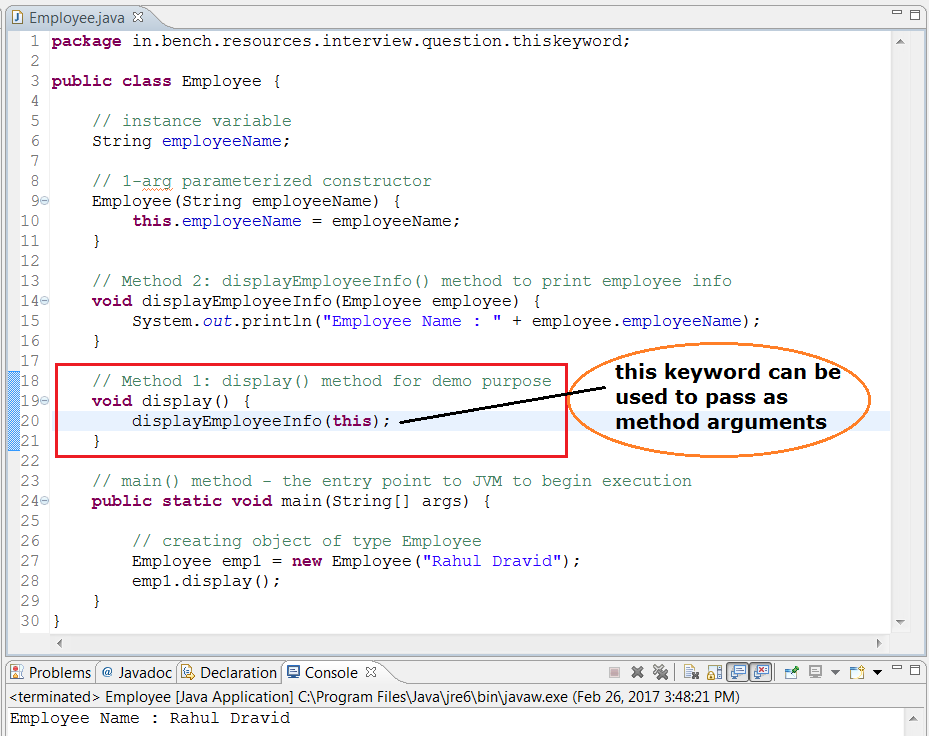
**Q) Whether it is possible to return this (this keyword) ?**

* Yes, this keyword can be used to return current class instance
* **Note:**To return *this* keyword (current instance) from method, we need to have return-type as current class-type; otherwise it won’t work

**[](http://www.benchresources.net/wp-content/uploads/2017/02/this-keyword-as-return-type-is-valid-9.png)**

**Q) Whether it is possible to pass this as method arguments ?**

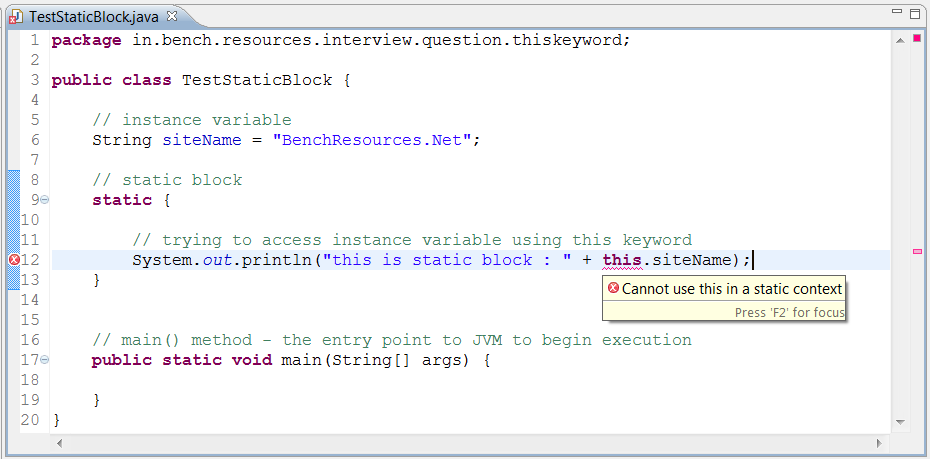
* Yes, *this*keyword can be used to pass as argument in method invocation
* Example, as shown in the below careen-capture

**[](http://www.benchresources.net/wp-content/uploads/2017/02/passing-this-as-method-argument-10.png)**

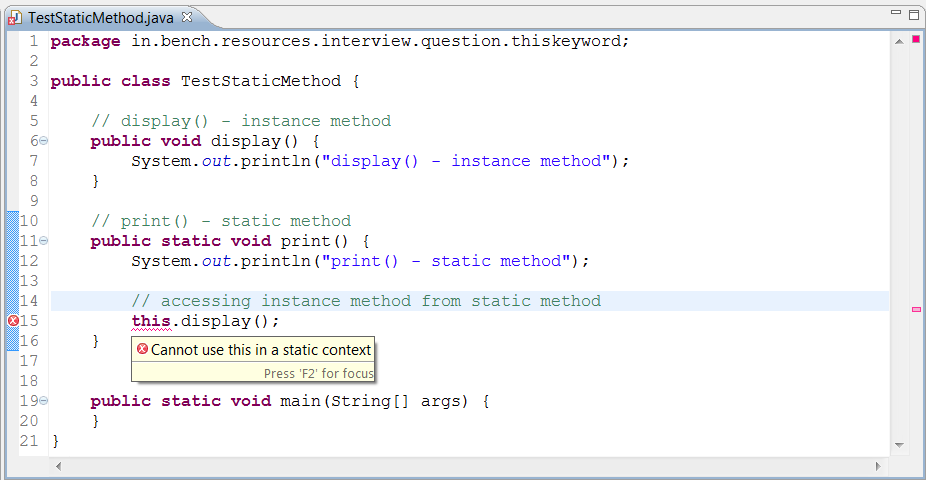
**Q) What will happen, if this keyword is used inside static blocks or static methods ?**

* Assigning references or accessing variables or invoking instance methods using ***this keyword*** from static contexts i.e.; within ***static block***or ***static methods*** results in ***compile-time error*** stating “***Cannot use this in a static context***”
* Let us see two example for these cases
* **Case 1:** accessing instance variable from static block
* **Case 2:** invoking instance method from static method
* Both cases results in ***compile-time error***with message stating “***Cannot use this in a static context***“

**Case 1:** accessing instance variable from static block

**[](http://www.benchresources.net/wp-content/uploads/2017/02/accessing-instance-variable-from-static-block-11a.png)**

**Case 2:** invoking instance method from static method

**[](http://www.benchresources.net/wp-content/uploads/2017/02/invoking-instance-method-from-static-method-11b.png)**

**Q1.  What are the common uses of "this" keyword in java ?**

Ans. "this" keyword is a reference to the current object and can be used for following -

1. Passing itself to another method.

2. Referring to the instance variable when local variable has the same name.

3. Calling another constructor in constructor chaining.

**Q2.  What is "this" keyword used for ?**

Ans. Used to represent an instance of the class in which it appears.

**Q3.  Difference Between this() and super() ?**

Ans.

1.this is a reference to the current object in which this keyword is used whereas super is a reference used to access members specific to the parent Class.

2.this is primarily used for accessing member variables if local variables have same name, for constructor chaining and for passing itself to some method whereas super is primarily used to initialize base class members within derived class constructor.

**Q4.  Can we use "this" within static method ? Why ?**

Ans. No. Even though "this" would mean a reference to current object id the method gets called using object reference but "this" would mean an ambiguity if the same static method gets called using Class name.