Java Nested Interface

An interface i.e. declared within another interface or class is known as nested interface. The nested interfaces are used to group related interfaces so that they can be easy to maintain. The nested interface must be referred by the outer interface or class. It can't be accessed directly.

Points to remember for nested interfaces

There are given some points that should be remembered by the java programmer.

* Nested interface must be public if it is declared inside the interface but it can have any access modifier if declared within the class.
* Nested interfaces are declared static implicitely.

Syntax of nested interface which is declared within the interface

1. **interface** interface\_name{
2. ...
3. **interface** nested\_interface\_name{
4. ...
5. }
6. }

Syntax of nested interface which is declared within the class

1. **class** class\_name{
2. ...
3. **interface** nested\_interface\_name{
4. ...
5. }
6. }

Example of nested interface which is declared within the interface

|  |
| --- |
| In this example, we are going to learn how to declare the nested interface and how we can access it. |

1. **interface** Showable{
2. **void** show();
3. **interface** Message{
4. **void** msg();
5. }
6. }
7. **class** TestNestedInterface1 **implements** Showable.Message{
8. **public** **void** msg(){System.out.println("Hello nested interface");}
10. **public** **static** **void** main(String args[]){
11. Showable.Message message=**new** TestNestedInterface1();//upcasting here
12. message.msg();
13. }
14. }

**[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=TestNestedInterface1" \t "_blank)**

[download the example of nested interface](https://static.javatpoint.com/src/nested/nestedinterface.zip)

Output:hello nested interface

|  |
| --- |
| As you can see in the above example, we are acessing the Message interface by its outer interface Showable because it cannot be accessed directly. It is just like almirah inside the room, we cannot access the almirah directly because we must enter the room first. In collection frameword, sun microsystem has provided a nested interface Entry. Entry is the subinterface of Map i.e. accessed by Map.Entry. |

Internal code generated by the java compiler for nested interface Message

|  |
| --- |
| The java compiler internally creates public and static interface as displayed below:. |

1. **public** **static** **interface** Showable$Message
2. {
3. **public** **abstract** **void** msg();
4. }

Example of nested interface which is declared within the class

|  |
| --- |
| Let's see how can we define an interface inside the class and how can we access it. |

1. **class** A{
2. **interface** Message{
3. **void** msg();
4. }
5. }
7. **class** TestNestedInterface2 **implements** A.Message{
8. **public** **void** msg(){System.out.println("Hello nested interface");}
10. **public** **static** **void** main(String args[]){
11. A.Message message=**new** TestNestedInterface2();//upcasting here
12. message.msg();
13. }
14. }

**[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=TestNestedInterface2" \t "_blank)**

Output:hello nested interface

Can we define a class inside the interface?

Yes, If we define a class inside the interface, java compiler creates a static nested class. Let's see how can we define a class within the interface:

1. **interface** M{
2. **class** A{}
3. }