Java Local inner class

A class i.e. created inside a method is called local inner class in java. If you want to invoke the methods of local inner class, you must instantiate this class inside the method.

Java local inner class example

1. **public** **class** localInner1{
2. **private** **int** data=30;//instance variable
3. **void** display(){
4. **class** Local{
5. **void** msg(){System.out.println(data);}
6. }
7. Local l=**new** Local();
8. l.msg();
9. }
10. **public** **static** **void** main(String args[]){
11. localInner1 obj=**new** localInner1();
12. obj.display();
13. }
14. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=localInner1)

Output:

30

Internal class generated by the compiler

In such case, compiler creates a class named Simple$1Local that have the reference of the outer class.

1. **import** java.io.PrintStream;
2. **class** localInner1$Local
3. {
4. **final** localInner1 **this**$0;
5. localInner1$Local()
6. {
7. **super**();
8. **this**$0 = Simple.**this**;
9. }
10. **void** msg()
11. {
12. System.out.println(localInner1.access$000(localInner1.**this**));
13. }
14. }

Rule: Local variable can't be private, public or protected.

Rules for Java Local Inner class

1) Local inner class cannot be invoked from outside the method.

2) Local inner class cannot access non-final local variable till JDK 1.7. Since JDK 1.8, it is possible to access the non-final local variable in local inner class.

Example of local inner class with local variable

1. **class** localInner2{
2. **private** **int** data=30;//instance variable
3. **void** display(){
4. **int** value=50;//local variable must be final till jdk 1.7 only
5. **class** Local{
6. **void** msg(){System.out.println(value);}
7. }
8. Local l=**new** Local();
9. l.msg();
10. }
11. **public** **static** **void** main(String args[]){
12. localInner2 obj=**new** localInner2();
13. obj.display();
14. }
15. }

**[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=localInner2" \t "_blank)**

Output:

50