Java Member inner class

A non-static class that is created inside a class but outside a method is called member inner class.

Syntax:

1. **class** Outer{
2. //code
3. **class** Inner{
4. //code
5. }
6. }

Java Member inner class example

In this example, we are creating msg() method in member inner class that is accessing the private data member of outer class.

1. **class** TestMemberOuter1{
2. **private** **int** data=30;
3. **class** Inner{
4. **void** msg(){System.out.println("data is "+data);}
5. }
6. **public** **static** **void** main(String args[]){
7. TestMemberOuter1 obj=**new** TestMemberOuter1();
8. TestMemberOuter1.Inner in=obj.**new** Inner();
9. in.msg();
10. }
11. }

[**Test it Now**](http://www.javatpoint.com/opr/test.jsp?filename=TestMemberOuter1)

Output:

data is 30

Internal working of Java member inner class

The java compiler creates two class files in case of inner class. The class file name of inner class is "Outer$Inner". If you want to instantiate inner class, you must have to create the instance of outer class. In such case, instance of inner class is created inside the instance of outer class.

Internal code generated by the compiler

The java compiler creates a class file named Outer$Inner in this case. The Member inner class have the reference of Outer class that is why it can access all the data members of Outer class including private.

1. **import** java.io.PrintStream;
2. **class** Outer$Inner
3. {
4. **final** Outer **this**$0;
5. Outer$Inner()
6. {   **super**();
7. **this**$0 = Outer.**this**;
8. }
9. **void** msg()
10. {
11. System.out.println((**new** StringBuilder()).append("data is ")
12. .append(Outer.access$000(Outer.**this**)).toString());
13. }
14. }