NESTED AND INNER CLASSES

Prepared by

Dr. Rajesh Kumar Ojha
Asst. Prof., CSE, Silicon University

1. To define a class within another class; is known as nested class.
2. If class B is defined within class A, then B is known to A, but not outside of A.
3. A nested class has access to the members, including private members, of the class in which it is nested.
4. However, the enclosing class does not have access to the members of the nested class.
5. Nested classes are used **to develop more readable and maintainable code** because it logically group classes and interfaces in one place only.

- Inner class is a part of nested class. **Non-static nested classes are known as inner classes.**
- There are two types of nested classes non-static and static nested classes. The non-static nested classes are also known as inner classes.

Type	Description
Member Inner Class	A class created within class and outside method.
Anonymous Inner Class	A class created for implementing interface or extending class. Its name is decided by the java compiler.
Local Inner Class	A class created within method.
Static Nested Class	A static class created within class.
Nested Interface	An interface created within class or interface.

Member Inner Class

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

```
class Outer{
 //code
 class Inner{
  //code
 }
}
```

```
class TestMemberOuter1{
 private int data=30;
 class Inner{
  void msg(){System.out.println("data is "+data);}
 }
 public static void main(String args[]){
  TestMemberOuter1 obj=new TestMemberOuter1();
  TestMemberOuter1.Inner in=obj.new Inner();
  in.msg();
 }
}
```

```
class Outer {
  int outer_x = 100;
  void test() {
    Inner inner = new Inner();
    inner.display();
  }

  // this is an inner class
  class Inner {
    void display() {
      System.out.println("display: outer_x = " + outer_x);
    }
  }
}

class InnerClassDemo {
  public static void main(String args[]) {
    Outer outer = new Outer();
    outer.test();
  }
}
```

Static Nested Class

Definition: A static class defined within another class

Characteristics:

1. Can access static members of the outer class
2. Cannot access non-static members directly

Example Code:

```
public class OuterClass {
    static int outerStatic = 10;
    static class StaticNestedClass {
        void display() {
            System.out.println("Outer static: " +
outerStatic);
        }
    }
}
```

Inner Class

Definition: A non-static class defined within another class

Characteristics:

1. Can access all members of the outer class
2. Requires an instance of the outer class to instantiate

Example Code:

```
public class OuterClass {
    int outerNonStatic = 20;
    class InnerClass {
        void display() {
            System.out.println("Outer non-static: " + outerNonStatic);
        }
    }
}
```

Local Inner Class

Definition: A class defined within a method of the outer class

Characteristics:

Can access final or effectively final variables of the enclosing method

Example Code:

```
public class OuterClass {
    void outerMethod() {
        final int localVar = 30;
        class LocalInnerClass {
            void display() {
                System.out.println("Local variable: " + localVar);
            }
        }
        LocalInnerClass localInner = new LocalInnerClass();
        localInner.display();
    }
}
```

Anonymous Inner Class

Definition: A class without a name defined within a method

Characteristics:

Used for implementing interfaces or extending classes on the fly

Example Code:

```
public class OuterClass {
    void outerMethod() {
        Runnable r = new Runnable() {
            public void run() {
                System.out.println("Anonymous Inner
Class");
            }
        };
        new Thread(r).start();
    }
}
```

Thank you