Nested Classes

## Definition:

The java programming language allows you to define a class with in another class. Such a class is called nested class.

//outer class

class OuterClass {

//inner class

class InnerClass {

}

}

## Types of Nested Classes:

Static Nested: In this, nested classes are declared by using the static keyword.

Non-Static Nested: These are also called inner classes and declared without using the static keyword.

A nested class can be declared private, public, protected.

## Why Nested Classes ?

To increase encapsulation

More readable & maintainable code.

It is a way of logically grouping the classes that are only used in one place.

## Static Nested Class

Nested classes that are declared with static keyword is called as static nested class. It can only access the static members of the outer class.

class OuterClass {

...

static class StaticNestedClass {

...

}

}

StaticNestedClassDemo.java

In the above program, we are accessing the static nested class code without creating an object for the outer class. We can also take the main() method inside the static nested class.

## Inner Class:

If a class declared inside another class directly, such type of class is called as inner class.

An inner class is always associated with an instance of its outer class.

Inside inner class we are not allowed to place any static declarations.

Within the inner class, we can access both static and non-static members of the outer class.

class OuterClass {

private int x;

//inner class

class InnerClass {

OuterClass.this.x = 22; // way to use this keyword to refer outer class property

}

}

InnerClassSample.java

Above program demonstrates accessing the inner class method from

1. Instance method of the outer class
2. Static method of outer class.

## Anonymous Inner Class

If we are not specifying any name for the inner class then we call that class as Anonymous inner class. This will allow us to declare and instantiate the class at the same time.

Will prefer these type of classes, if we are going to use them only once.

Syntax: HelloWorld helloWorldObj = new HelloWorld() {

// properties

// override or implement the relevant method

}**;**

**Note**: HelloWorld may be an interface or class.

Anonymous class expression is part of a statement so it must ends with semicolon.

However, you cannot declare constructors in an anonymous class.

AnonymousInnerClass.java

In the above sample, Employee is a class and we are creating an anonymous inner class for that.

## Method Local Inner Class

If we want to specify any method specific behaviour then we will prefer method local inner classes.

i.e. Method local class is a nested class declared inside a method.

If we declare the method local inner class inside the instance method, then we can access both static and non-static members of the outer class.

If we declare the method local inner class inside the static method, then we can access only the static members of the outer class.

From this, we are not allowed to use the local variables of its enclosing method unless the local variables are defined as final.

class OuterClass {

private void method1(){

//inner class

class InnerClass {

}

}

}

MethodLocalInnerClasssDemo.java

Above method local inner class, demonstrates the ways to access final and non-final variables of the method.