Java Inner Class

Java inner class or nested class is a class i.e. declared inside the class or interface.

We use inner classes to logically group classes and interfaces in one place so that it can be more readable and maintainable.

Additionally, it can access all the members of outer class including private data members and methods.

Syntax

```
class Outer_class{

 //code

class Inner_class{

  //code

}

}
```

Advantage of java inner classes

There are basically three advantages of inner classes in java. They are as follows:

1) Nested classes represent a special type of relationship that is it can access all the members (data members and methods) of outer class including private.

2) Nested classes are used to develop more readable and maintainable code because it logically group classes and interfaces in one place only.

3) Code Optimization: It requires less code to write.

Difference between nested class and inner class in Java

Inner class is a part of nested class. Non-static nested classes are known as inner classes.

Types of Nested 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.

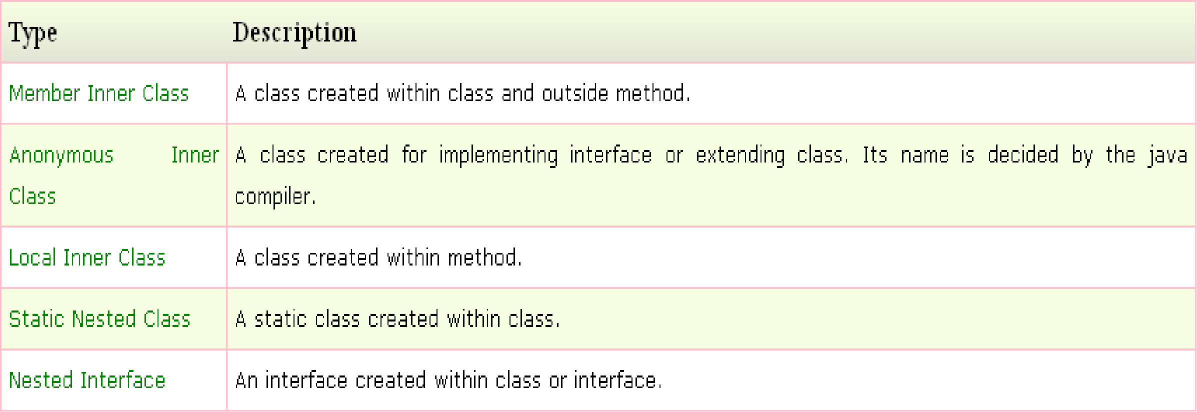
1. Non-static nested class(inner class)

   a)Member inner class

   b)Anonymous inner class

c)Local inner class

2. Static nested class



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.

Java 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
 }  }
```

Following example demonstrates a nested class.

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

in.msg();

 }

}
```

Note:The java compiler creates two class files in case of inner class. The class file name of inner class is "TestMemberOuter$Inner".

Java Anonymous inner class

A class that has no name is known as anonymous inner class in java.

It should be used if you have to override method of class or interface. Java Anonymous inner class can be created by two ways:

1. Class (may be abstract or concrete).
2. Interface

Java anonymous inner class example using class

```
abstract class Person{

abstract void eat();

}

class TestAnonymousInner{

public static void main(String args[]){

  Person p=new Person(){

        void eat()

        {

        System.out.println("nice fruits");

        }

  };

p.eat();

 }

}
```

1. A class is created but its name is decided by the compiler which extends the Person class and provides the implementation of the eat() method.

2. An object of Anonymous class is created that is referred by “p” reference variable of Person type.

Java anonymous inner class example using interface

```
interface Eatable{
void eat();
}
class TestAnnonymousInner1{
public static void main(String args[]){
 Eatable e=new Eatable(){
public void eat(){System.out.println("nice fruits");}
 };
e.eat();
 }
}
```

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.

Or we can say **Inner class** can be declared within a method of an outer class.

Example

```
public class TestLocalInner{
private int data=30;//instance variable
void display(){
class Local{
void msg(){System.out.println(data);}
  }
  Local l=new Local();
l.msg();
```

```
 }
public static void main(String args[]){
TestLocalInner obj=new TestLocalInner();
obj.display();
 } }
```

Note: Local inner class cannot be invoked from outside the method.

Java static nested class

A static class i.e. created inside a class is called static nested class in java. It cannot access non-static data members and methods. It can be accessed by outer class name.

- It can access static data members of outer class including private.
- Static nested class cannot access non-static (instance) data member or method.

Example

```
classTestOuter{
static int data=30;
static class Inner{
void msg(){System.out.println("data is "+data);}
  }
public static void main(String args[]){
TestOuter.Innerobj=new TestOuter1.Inner();
obj.msg();
  }
}
```

Java static nested class example with static method

If you have the static member inside static nested class, you don't need to create instance of static nested class.

```
classTestOuter{
static int data=30;
static class Inner{
```

```
static void msg(){System.out.println("data is "+data);}
  }
public static void main(String args[]){
TestOuter.Inner.msg();//no need to create the instance of static nested class
  }
}
```