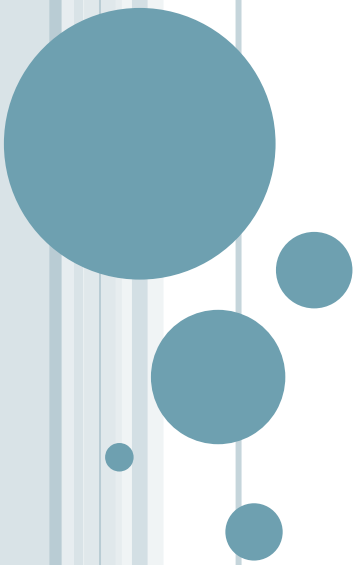
NESTED CLASSES



NESTED CLASSES

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

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

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

Syntax of Inner class

```
class Java_Outer_class{
 //code
 class Java_Inner_class{
  //code
 }
}
```

NESTED CLASSES

Advantage of Java inner classes

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

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

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

Code Optimization: It requires less code to write.

Need of Java Inner class

Sometimes users need to program a class in such a way so that no other class can access it. Therefore, it would be better if you include it within other classes.

If all the class objects are a part of the outer object then it is easier to nest that class inside the outer class. That way all the outer class can access all the objects of the inner class.

NESTED CLASSES

Difference between nested class and inner class in Java

An inner class is a part of a 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.

- Non-static nested class (inner class)
  - Member inner class
  - Anonymous inner class
  - Local inner class
- Static nested class

NESTED CLASSES

Type	Description
Member Inner Class	A class created within class and outside method.
Anonymous Inner Class	A class created for implementing an interface or extending class. The java compiler decides its name.
Local Inner Class	A class was created within the method.
Static Nested Class	A static class was created within the class.
Nested Interface	An interface created within class or interface.

NESTED CLASSES

- **Java Member Inner class**
- A non-static class that is created inside a class but outside a method is called member inner class. It is also known as a regular inner class. It can be declared with access modifiers like public, default, private, and protected.
- Java Member Inner Class Example

```
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();

    }

   }
```

NESTED CLASSES

- **Internal working of Java member inner class**
- The java compiler creates two class files in the case of the inner class. The class file name of the inner class is "Outer$Inner". If you want to instantiate the inner class, you must have to create the instance of the outer class.

NESTED CLASSES

- **Java Anonymous inner class**
- Java anonymous inner class is an inner class without a name and for which only a single object is created. An anonymous inner class can be useful when making an instance of an object with certain "extras" such as overloading methods of a class or interface, without having to actually subclass a class.
- In simple words, a class that has no name is known as an anonymous inner class in Java. It should be used if you have to override a method of class or interface. Java Anonymous inner class can be created in two ways:

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

NESTED CLASSES

```
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();
 }
}
```

O/p nice fruits

NESTED CLASSES

- **Internal working of given code**

- 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.
- An object of the Anonymous class is created that is referred to by 'p,' a reference variable of Person type.

NESTED CLASSES

- **Java Local inner class**

- A class i.e., created inside a method, is called local inner class in java. Local Inner Classes are the inner classes that are defined inside a block. Generally, this block is a method body. Sometimes this block can be a for loop, or an if clause. Local Inner classes are not a member of any enclosing classes. They belong to the block they are defined within, due to which local inner classes cannot have any access modifiers associated with them. However, they can be marked as final or abstract. These classes have access to the fields of the class enclosing it.
- If you want to invoke the methods of the local inner class, you must instantiate this class inside the method.

NESTED CLASSES

```
public class localInner1{
 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[]){
  localInner1 obj=new localInner1();
  obj.display();
 }
} o/p: 30
```

NESTED CLASSES

- **Java static nested class**
- A static class is a class that is created inside a class, is called a 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 the outer class, including private.
- The static nested class cannot access non-static (instance) data members

NESTED CLASSES

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

o/p: 30

SUMMARY OF NESTED CLASSES

A class defined within another class is called a nested class. Like other members of a class, a nested class can be declared static or not. A nonstatic nested class is called an inner class. An instance of an inner class can exist only within an instance of its enclosing class and has access to its enclosing class's members even if they are declared private.

CONT..

The following table shows the types of nested classes:

Types of Nested Classes		
Type	**Scope**	**Inner**
static nested class	member	no
inner [non-static] class	member	yes
local class	local	yes
anonymous class	only the point where it is defined	yes