Java Inner 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

 }

}

### Advantage of Java inner classes

1.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.

2.**Code Optimization**: It requires less code to write

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)
  1. Member inner class
  2. Anonymous inner class
  3. Local inner class
* Static nested class

|  |  |
| --- | --- |
| **Type** | **Description** |
| [Member Inner Class](https://www.javatpoint.com/member-inner-class) | A class created within class and outside method. |
| [Anonymous Inner Class](https://www.javatpoint.com/anonymous-inner-class) | A class created for implementing an interface or extending class. The java compiler decides its name. |
| [Local Inner Class](https://www.javatpoint.com/local-inner-class) | A class was created within the method. |
| [Static Nested Class](https://www.javatpoint.com/static-nested-class) | A static class was created within the class. |

# 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**.

**Syntax:**

class Outer{

//code

class Inner{

//code

}

}

**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();

} }

# Java Anonymous inner class

Java anonymous inner class is an inner class without a name and for which only a single object is created.

 Java Anonymous inner class can be created in two ways:

**1.Class (may be abstract or concrete).**

**2.Interface**

**Class (may be abstract or concrete).**

**abstract** **class** student{

**abstract** **void** message();

}

**class** Anonymous{

**public** **static** **void** main(String args[]){

  student p=**new** student(){

**void** message(){System.out.println("nice fruits");}

  };

  p.eat();

 }

}

**2.Interface**

**interface** A{

**void** display();

}

**class** TestAnnonymousInner1{

**public** **static** **void** main(String args[]){

 A e=**new** A(){

**public** **void** eat(){System.out.println("welcome");}

 };

 e.display();

 }

}

# 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.

**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();

 }

}

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 or

**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();

  }

}