**Inner Classes in Java**

This document explains the concept of **Inner Classes** in Java using the examples and key points derived from your notes.

**1. Types of Inner Classes in Java**

Java provides four types of inner classes:

1. **Instance Inner Class** (Non-Static Inner Class)
2. **Static Inner Class**
3. **Local Inner Class**
4. **Anonymous Inner Class**

We will explore each type in detail with examples and key highlights.

**2. Instance Inner Class**

An **Instance Inner Class** is a class defined inside another class without using the static keyword. It is associated with an object of the outer class.

**Example**

public class A {

class B {

void display() {

System.out.println("Instance Inner Class");

}

}

public static void main(String[] args) {

A a = new A(); // Create outer class object

A.B b = a.new B(); // Create inner class object

b.display(); // Access inner class method

}

}

**Key Points**

* **Compilation Output**: This code generates two .class files:
  + A.class (Outer class)
  + A$B.class (Inner class)
* **Naming Convention**: OuterClass$InnerClass for the inner class.
* **Accessing Outer Class Members**:
* public class A {
* int i = 10;
* static int j = 30;
* class B {
* void display() {
* System.out.println(i); // Access outer non-static member
* System.out.println(j); // Access outer static member
* }
* }
* }

**3. Static Inner Class**

A **Static Inner Class** is defined using the static keyword. It does not require an instance of the outer class to be created.

**Example**

public class X {

static class Y {

void display() {

System.out.println("Static Inner Class");

}

}

public static void main(String[] args) {

X.Y y = new X.Y(); // Create static inner class object directly

y.display();

}

}

**Key Points**

* **No Outer Class Instance Needed**: Static inner classes can be instantiated without creating an outer class object.
* **Static Members**: Static inner classes can declare static and non-static members.
* **Accessing Outer Class Members**: Static inner classes cannot access non-static members of the outer class directly.

**4. Local Inner Class**

A **Local Inner Class** is defined within a method or block, and its scope is limited to the enclosing block.

**Example**

public class P {

void m1() {

class Q {

void display() {

System.out.println("Local Inner Class in m1()");

}

}

Q q = new Q(); // Create local inner class instance

q.display();

}

public static void main(String[] args) {

new P().m1();

}

}

**Key Points**

* **Scope**: Local inner classes are accessible only within the method or block where they are defined.
* **Static Members**: Local inner classes cannot have static declarations.
* **Generated Class File**: Naming follows the format OuterClass$Index+LocalInnerClass.

**5. Anonymous Inner Class**

An **Anonymous Inner Class** is a class defined without a name, typically used for:

* Implementing abstract classes
* Overriding methods of a class

**Example 1: Extending a Class**

class Person {

void m1() {

System.out.println("Person - m1");

}

}

public class InnerClassDemo {

public static void main(String[] args) {

Person p = new Person() {

int i = 10;

void m1() {

System.out.println("Anonymous Inner Class - m1");

System.out.println(i);

}

};

p.m1();

}

}

**Example 2: Abstract Class Implementation**

abstract class Vehicle {

abstract void addBody();

}

public class InnerClassDemo {

public static void main(String[] args) {

Vehicle car = new Vehicle() {

@Override

void addBody() {

System.out.println("Fixed body for anonymous car");

}

};

car.addBody();

}

}

**Key Points**

* **Purpose**: Anonymous inner classes simplify code when you need an inner class only once.
* **Super Class Members**: You can override methods of the superclass but cannot add new methods accessible via the reference variable.
* **Generated Class File**: The name of the .class file follows the pattern OuterClass$Index.class.

**6. Summary Table**

| **Type of Inner Class** | **Defined In** | **Static Allowed?** | **Scope** |
| --- | --- | --- | --- |
| **Instance Inner Class** | Inside a class (non-static) | No | Entire Outer Class |
| **Static Inner Class** | Inside a class (static keyword) | Yes | Entire Outer Class |
| **Local Inner Class** | Inside a method/block | No | Enclosing Method/Block |
| **Anonymous Inner Class** | Inline (no class name) | No | Immediate instantiation |

**7. Compilation Notes**

* **Instance Inner Class**: Generates OuterClass$InnerClassName.class
* **Static Inner Class**: Generates OuterClass$StaticInnerClassName.class
* **Local Inner Class**: Generates OuterClass$Index+LocalInnerClassName.class
* **Anonymous Inner Class**: Generates OuterClass$Index.class

This document captures the essence of inner classes in Java, their syntax, behavior, and key highlights, along with real-world examples based on your notes.