Here are detailed and structured notes from the uploaded document:

Inner Class in Java

Introduction to Inner Classes

- **Definition**: An inner class is a class defined inside another class. The class that contains the inner class is known as the outer or enclosing class.
- **Scope**: The scope of the inner class is tied to the outer class. An inner class cannot exist independently without its outer class.

Inner Class vs Nested Class

- **Nested Classes**: A class declared inside another class is called a nested class.
- **Inner Classes**: A non-static nested class in Java is called an inner class. It has access to the outer class's members (variables and methods).

Syntax of Inner Class

```
access_modifier class OuterClassName {
    access_modifier class InnerClassName {
        // Members of the inner class
    }
    // Other members of the outer class
}
```

Example:

```
public class Outer {
    public class Inner {
        // Inner class code
    }
}
```

Features of Inner Class

1. **Naming Restrictions**: An inner class cannot have the same name as the outer class, though members of both classes can have the same name.
2. **Access to Outer Class Members**: An inner class can access all members of the outer class, including private ones.
3. **Object Creation**: Inner class objects are created with an existing outer class object.
4. **Object Scope**: The inner class exists only as long as the outer class exists.
5. **Encapsulation**: Inner classes provide better encapsulation by hiding their implementation details within the outer class.

Instantiation of Inner Class

To instantiate an inner class:

```
OuterClass.InnerClass innerObject = outerObject.new InnerClass();
```

Example:

```
Outer outer = new Outer();
Outer.Inner inner = outer.new Inner();
```

Types of Inner Classes in Java

1. **Regular Inner Class**: Defined inside a class but outside any methods.
2. **Method Local Inner Class**: Defined inside a method of the outer class.
3. **Anonymous Inner Class**: A nameless inner class used when only one instance is needed.
4. **Static Nested Class**: A nested class defined with the static modifier, which can access only static members of the outer class.

Advantages of Inner Class

1. **Logical Grouping**: Inner classes help group classes and interfaces that logically belong together.
2. **Access to Outer Class Members**: They can directly access all members of the outer class, including private members.
3. **Better Readability**: Reduces the need for extra code and increases maintainability.
4. **Encapsulation**: Inner classes help achieve tighter encapsulation, reducing code complexity.

How to Instantiate a Member Inner Class

1. Create an instance of the outer class.
2. Use that instance to create an instance of the inner class. Example:

   ```
   Outer outer = new Outer();
   Outer.Inner inner = outer.new Inner();
   ```

   Alternatively:

   ```
   Outer.Inner inner = new Outer().new Inner();
   ```

Anonymous Inner Class

- **Definition**: An anonymous inner class is a class that is declared without a name. It's often used when you need only one instance of a class.
- **Syntax**:

  ```
  SomeClass obj = new SomeClass() {
      // Implementation of methods
  };
  ```

Static Nested Class

- **Definition**: A static nested class is a class that is declared static within an outer class. Unlike inner classes, it can exist independently of the outer class instance.

- **Example**:

```
public class Outer {
    public static class StaticNested {
        // Body of the StaticNested class
    }
}
```

- **Main Method**: A static nested class can contain a `main` method and can be run independently.
- **Object Creation**: Static nested classes can be instantiated without an instance of the outer class.

```
Outer.StaticNested nestedObj = new Outer.StaticNested();
```

FAQs

1. **Can you declare a class inside a class?**
    - Yes, classes can be declared inside other classes.
2. **Can you declare a class inside an interface?**
    - Yes, it’s valid to declare classes inside interfaces.
3. **Can you declare an interface inside a class?**
    - Yes, this is valid in Java.
4. **Can you declare an interface inside another interface?**
    - Yes, interfaces can be declared inside other interfaces.

Important Points

- **Class & Interface Combinations**: Java allows flexible nesting of classes and interfaces.
- **Instance of Static Nested Class**: Can be created independently of the outer class instance.

These notes provide a thorough overview of inner classes in Java, including their types, advantages, syntax, and examples. Let me know if you need further details or clarifications!