Inner Class in 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.

Inner Class in Java

Syntax of Inner class :

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

Inner Class in Java

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.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.

Inner Class in Java

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.

Inner Class in Java

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.

Inner Class in Java

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**

Inner Class in Java

Types of 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.

Inner Class in Java

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.

Syntax:

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

Inner Class in Java

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();
 }
}
```

Inner Class in Java

How to instantiate Member Inner class in Java?

An object or instance of a member's inner class always exists within an object of its outer class. The new operator is used to create the object of member inner class with slightly different syntax.

The general form of syntax to create an object of the member inner class is as follows:

Inner Class in Java

How to instantiate Member Inner class in Java?

Syntax:

```
OuterClassReference.new MemberInnerClassConstructor();
```

Example:

```
obj.new Inner();
```

Here, OuterClassReference is the reference of the outer class followed by a dot which is followed by the new operator.

Inner Class in Java

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. In such a case, an instance of inner class is created inside the instance of the outer class.

Inner Class in Java

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:

Class (may be abstract or concrete).
Interface

Inner Class in Java

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.

Inner Class in Java

Java local inner class example :

```
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();   }  }
```

Inner Class in Java

Java local inner class example :

```
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();   }  }
```

Inner Class in Java

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

Inner Class in Java

Java static nested class example with instance method :

```
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();
    }
  }
```

Inner Class in Java

Internal class generated by the compiler :

```
import java.io.PrintStream;
  static class TestOuter1$Inner
  {
  TestOuter1$Inner(){}
  void msg(){
  System.out.println((new StringBuilder()).append("data is
")
  .append(TestOuter1.data).toString());
  }
  }
```

Inner Class in Java

Assignment Question :

1. What is the internal code generated by the compiler for member inner class?
2. What are the two ways to create an anonymous inner class?
3. Can we access the non-final local variable inside the local inner class?
4. How to access the static nested class?
5. Can we define an interface within the class?
6. Can we define a class within the interface?