Inner Classes or Nested Classes, Types of nested classes and examples,non-static inner classes and static inner classes

-K.L.Madhavi

Nested Classes or Inner 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.
- 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
    }
}
```

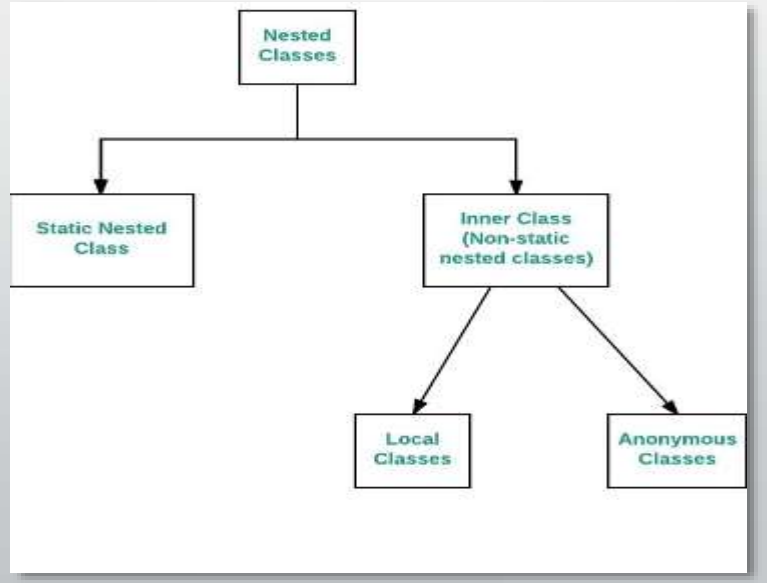
Advantage of Java inner classes

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.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.

- An inner class is a part of a nested class. Non-static nested classes are known as inner classes.
- Nested classes are divided into two categories:
  - **static nested class:** Nested classes that are declared *static* are called static nested classes.
  - **inner class:** An inner class is a non-static nested class.



Static Nested Class

In Java, a static nested class refers to a class defined within another class using the static keyword. It is distinct from non-static nested classes (inner classes) in that it does not depend on any instance of the outer class and can be accessed independently. It can be accessed and instantiated without requiring an outer class instance.

Static Nested Class Example Program

```
import java.io.*;
import java.util.*;

class ClassA {
    static int x = 10;
    int y = 20;
    private static int z = 30;
    //static nested class
    static class ClassB {
        void get()
        {

            System.out.println("x: " + x);
            System.out.println("z: " + z);
        }
    }
}
class SNestClass {
     public static void main(String[] args)
    {       // Creating object of  static nested class in main()
            ClassA.ClassB ob1 = new ClassA.ClassB();
             ob1.get();
    }
}
```

Output:

x: 10

z: 30

Non-Static (Inner) Nested Class

A non-static nested class in Java, also known as an inner class, is a class that is defined inside another class and is not marked with the static keyword. In Java, non-static nested classes, also known as inner classes, are intricately linked to an instance of the outer class, granting them direct access to its instance members. Inner classes can refer to and modify the instance variables and methods of the outer class without the need for explicit references. They provide a way to logically group classes together and enable them to interact more closely with each other.

Non-Static Nested Class Example Program

```
import java.io.*;
class ClassA {
    static int x = 10;
    int y = 20;
    public int z = 30;

    class ClassB {
        void get()
        {
            System.out.println("x: " + x);
            System.out.println("y: " + y);
            System.out.println("z: " + z);
        }
    }
}
class NonSNestClass {
    public static void main(String[] args)
    {   ClassA ob1 = new ClassA();
        // Creating object of non-static nested class in main()
        ClassA.ClassB ob2 = ob1.new ClassB();
         ob2.get();
    }
}
```

Output:

x: 10
y: 20
z: 30

Differences between Static Nested Class and Non-Static Nested Class:

Static Nested Class	Non-Static (Inner) Class
Can be instantiated independently.	Requires an instance of the outer class.
Independent of any instance of the outer class.	Tightly associated with an instance of the outer class.
Can access static members directly.	Can access both static and non-static members.
Cannot access instance fields directly.	Can access instance fields directly.
Cannot invoke non-static methods.	Can invoke both static and non-static methods.
Cannot access non-static nested classes.	Can access both static and non-static nested classes.