

## Nested Classes

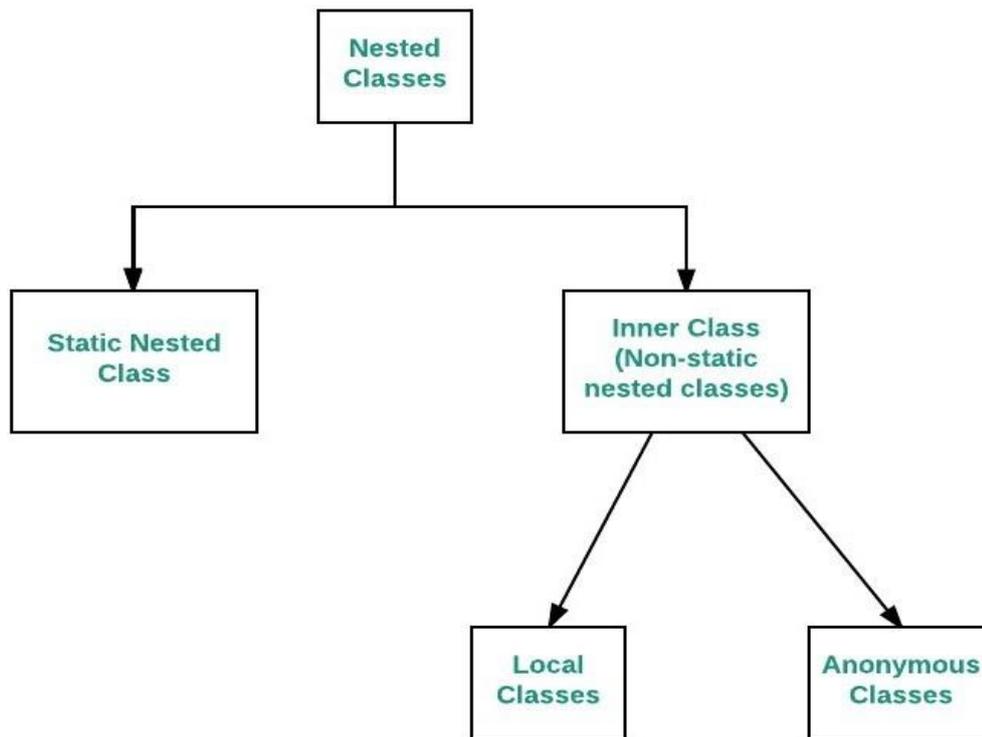
The Java programming language allows you to define a class within another class which is called nested class.

The following java segment illustrates the nested class declaration in java:

### Example 1

```
public class OuterClass {  
    ...  
    public class NestedClass {  
        ...  
    }  
}
```

**Very important note (Terminology):** Nested classes are divided into two categories:



**static and non-static nested classes.**

Nested classes that are declared static are simply called **static nested classes**. Non-static nested classes are called **inner classes**.

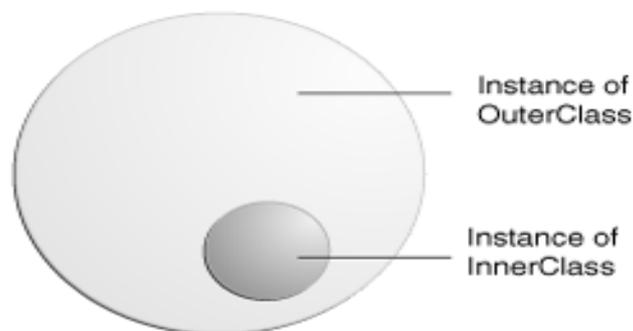
### Inner Classes

The inner class is known as a *member class*, and is just another class component in the same way that constants, variables, and methods are also class components. As with instance methods and variables, an inner class is associated with an instance of its enclosing class and has direct access to that object's methods and fields. Also, because an inner class is associated with an instance, it cannot define any static members itself. Objects that are instances of an inner class exist *within* an instance of the outer class.

#### *Access Modifiers*

You can use the same modifiers for inner classes that you use for other members of the outer class. For example, you can use the access specifiers — private, public, and protected — to restrict access to inner classes, just as you do to other class members.

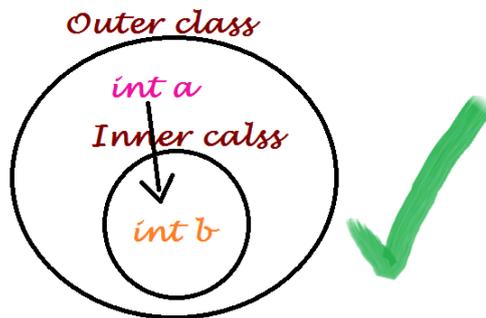
An instance of InnerClass can exist only within an instance of OuterClass and has direct access to the methods and fields of its enclosing instance. The next figure illustrates this idea.



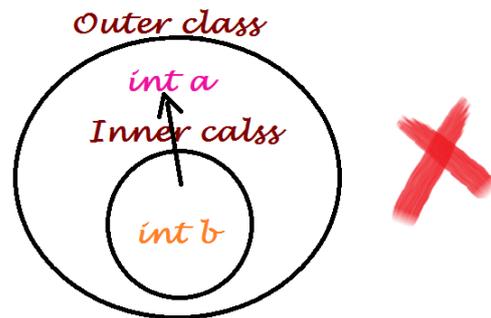
#### **An Instance of InnerClass Exists Within an Instance of OuterClass**

Notice from the following example that in order to access the fields of the member class, it has been necessary to create an instance of the class Center. The instance of the member class may then be used in the enclosing class.

## Scope of variables in nested classes.



We can access outer class variables in inner class



We cannot access inner class variables in outer class.

### Example 2 (Structure)

```
public class RoundShape
{
    public class Center{
        private int x,y;
        Center(){ }
    }

    private Center C = new Center();
    public float radiusOfCircle;
    .. }
}
```

## LEC20 OOP 2018-2019

### Example 3 (Trace)

What is the output of the following program:

```
public class outclass
{
    private int x;
    inner in=new inner();
    public void set()
    {
        x=100;
        in.set();
    }

    public void print()
    {
        System.out.println("x="+x);
        System.out.println("y="+in.get());
    }
}

public class inner
{
    private int y;
    public int get(){return (y);}
    public void set()
    {y=200;}
    public void set(int a){y=a;}
    public void print(){
        System.out.println("y="+y);}
}

public class Main {
    public static void main(String[] args) {

        outclass o= new outclass();
        o.set();
        o.print();

        outclass.inner i1=o.new inner();
        outclass.inner i2=new outclass().new inner();

        i1.set(8);    i1.print();
        i2.set(5);    i2.print();
    }
}
```

The output will be:

```
x=100
y=200
y=8
y=5
```

## LEC20 OOP 2018-2019

### Example 4 (Trace)

What is the output of the following java program?

```
public class Test {  
  
    public int num = 25;  
  
    public int getNum()  
    { return num; }  
  
    public class MyInner {  
        public int num = 38;  
  
        public int getNum() { return num; }  
    }  
}  
  
public class Main{  
  
    public static void main(String[] args) {  
        Test abt = new Test();  
  
        //OuterClass.InnerClass innerObject = outerObject.new InnerClass();  
  
        Test.MyInner minn =abt.new MyInner();  
        System.out.println(abt.num);  
        System.out.println(minn.num);  
    }  
}  
The output will be  
25  
38
```

### Static Nested Classes

A **static nested class** is a regular class defined inside of a package level class or inside of another static nested class. They are actually defined inside the body of the parent class, not only in the same file. As with any high level facility offered by a programming language it can be of real help in structuring clear programs or it can be just the opposite of this when abused.

**Static nested class** facts:

- is defined as a static member of the parent class
- accepts all accessibility modifiers
- it is NOT linked to an outer instance (it can live independently)
- has direct access to static members of the parent class regardless of the access modifiers declared in the parent class
- has direct access to all members of an instance of the parent class regardless of the access modifiers declared in the parent class

Here is a brief example of how nested classes are declared and how they access members of the parent classes.

#### Example 5

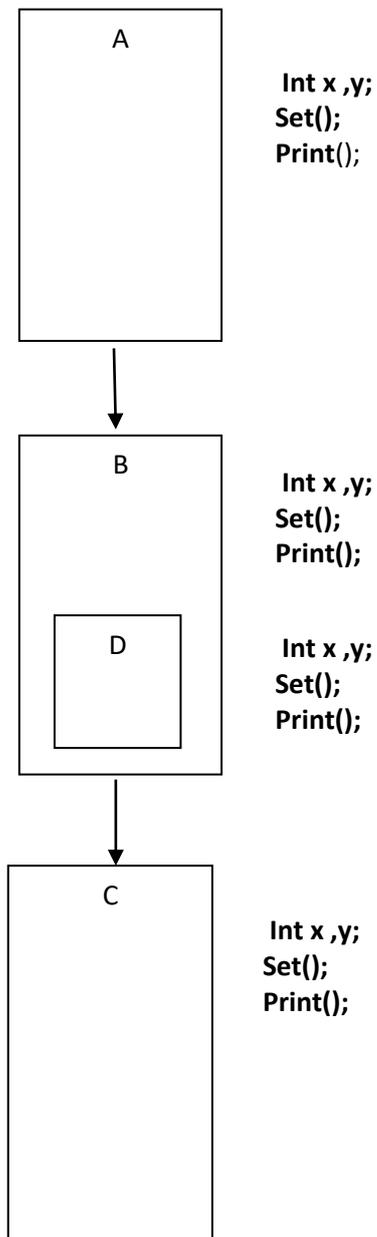
```
public class Top{  
    private static int staticCounter = 0;  
    private int nestedCounter = 0;  
  
    public static class Nested1 {  
        private static int staticCounter = 0;  
        private int nestedCounter = 0;  
  
        public static class Nested2 {  
            public Nested2(Top t, Top.Nested1 tn1) {  
                Top.staticCounter++;  
                t.nestedCounter++;  
                Top.Nested1.staticCounter++;  
                tn1.nestedCounter++; } }  
  
        public Nested1(Top t) {  
            Top.staticCounter++;  
            t.nestedCounter++; }  
  
        public String toString() {  
            return  
            getClass().getName() + ".nestedCounter: " + nestedCounter +
```

```
System.getProperty("line.separator") +  
getClass().getName() + ".staticCounter: " + staticCounter;  
}}  
public String toString() {  
return  
getClass().getName() + ".nestedCounter: " + nestedCounter +  
System.getProperty("line.separator") +  
getClass().getName() + ".staticCounter: " + staticCounter;  
}  
public static void main(String[] args) {  
Top t = new Top();  
Top.Nested1 nested1 = new Top.Nested1(t);  
Top.Nested1.Nested2 nested2 = new Top.Nested1.Nested2(t, nested1);  
System.out.println(t);  
System.out.println(nested1);  
}
```

## LEC20 OOP 2018-2019

### LAB:

Write a complete program to define the following classes:



In class main find the summation of all x's and y's