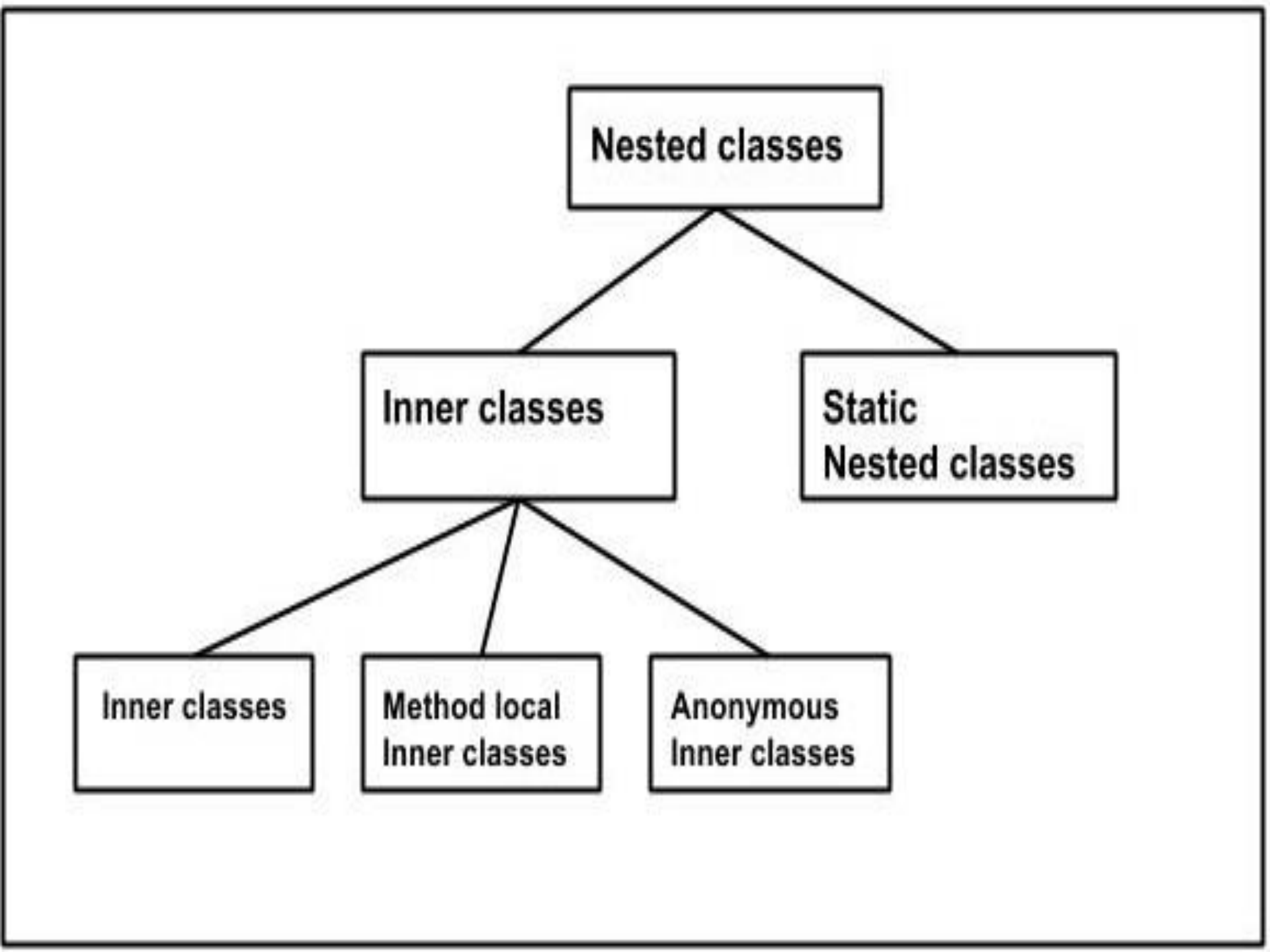
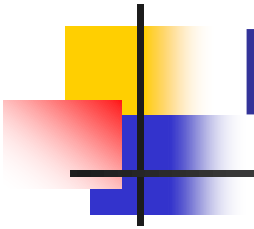
03. Class concepts – (Java)

Class fundamentals
Methods
Constructors
Inner Classes

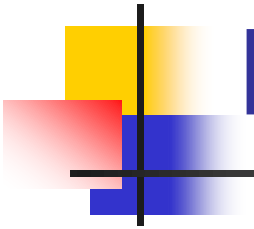

Nested classes
Inner classes
Static
Nested classes
Inner classes
Method local
Inner classes
Anonymous
Inner classes



Nested Classes

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

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

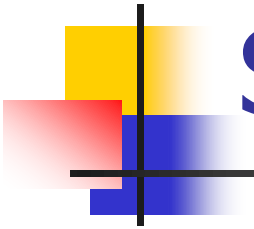


Nested Classes

- Nested classes are divided into two categories: static and non-static. Nested classes that are declared static are simply called *static nested classes*. Non-static nested classes are called *inner classes*

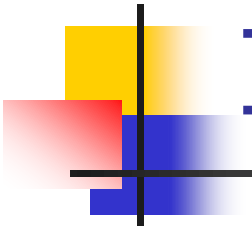
Nested Classes

```
class OuterClass {
        ...
        static class StaticNestedClass {
        ...
         }
        class InnerClass {
         ...
         }
}
```



Static Nested Classes

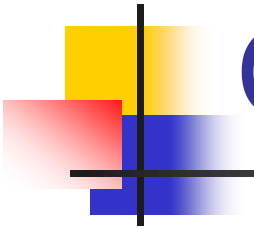
- As with class methods and variables, a static nested class is associated with its outer class. And like static class methods, a static nested class cannot refer directly to instance variables or methods defined in its enclosing class — it can use them only through an object reference.



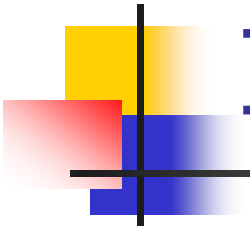
Inner Classes

- 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.

Local and Anonymous Inner Classes



- There are two additional types of inner classes. You can declare an inner class within the body of a method. Such a class is known as a *local inner class*. You can also declare an inner class within the body of a method without naming it. These classes are known as *anonymous inner classes*.

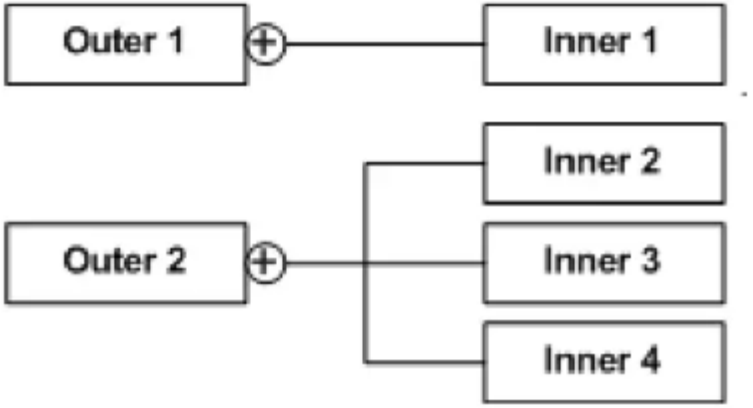


Inner Classes

- Advantage of Inner Classes
  - 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.
  - Nested classes are used **to develop more readable and maintainable code** because it logically group classes and interfaces in one place only.
  - **Code Optimization**: It requires less code to write.

Inner Classes - UML

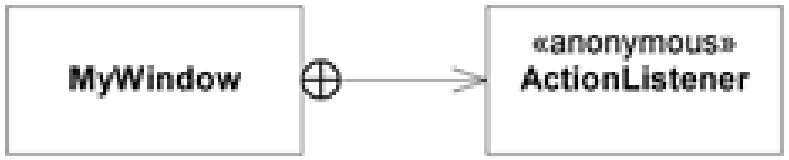
Nested class in UML (for any language) can be represented as:



Here

1. Class Inner1 is nested inside the outer class Outer 1
2. Classes Inner2, Inner3, Inner4 classes are nested inside Outer2

Inner Classes - UML




```
public class Window {
  public void f() {
    ActionListener l =
      new ActionListener() {
        // implementation
      };
  }
}
```