**Pencil And Paper For Lesson 5**

1. **Situation: Your method myMethod() will make use of the inner class several times, but the inner class will not be needed for any other purpose**

In this case we would use: (ii) Make it a local inner class, defined within myMethod(). Because a local inner class is defined precisely where it is needed. This makes it easier to maintain, and makes code easier to follow.

1. **Situation. Your method myMethod() will use the inner class just once, and it will not be referred to again; also, users of myMethod will make relatively few calls to myMethod()**

In this case we would use: (iii)Make it an **anonymous inner class,** defined within myMethod(). Because it can be used in place of local inner classes without creating an artificial auxiliary method and provides the same strong encapsulation as local inner classes.

1. **Situation. Your method myMethod() will be accessed and repeatedly called in a loop from an instance of another class.**

In this case we would use: (i) Make it a member inner class. Because this behavior can be costly and it is usually better to move a member inner class and instantiate it just once, if its values need to take on different values as a loop executes, these can be set using setter methods.