Inner Classes refer to any class that is inside of another class. We often times call the inner class a Nested Class and the outside class an Outer Class. When it comes to these inner classes, we can break them down into both static and non-static inner classes. A static Nested Class cannot directly refer to an instance variable or method in the enclosing class. It can only use object reference. An inner class that is set to private cannot be accessed by the outer class. This acts a sort of security measure for access. There are also Anonymous Inner Classes and these do not have a name. They are declared and instantiated at the same time. The most common use for an Anonymous Class is when we want to override the method of a class or interface (tutorialspoint.com)

According to Oracle, three of the most common reasons to use Nested Classes is it is a way of grouping classes, it increases encapsulation and can lead to more readable and maintainable code. When it comes to grouping, if one class is only really useful to one other class, it may make sense to create a Nested Class. When it comes to encapsulation, “Consider two top-level classes, A and B, where B needs access to members of A that would otherwise be declared private. By hiding class B within class A, A's members can be declared private and B can access them. In addition, B itself can be hidden from the outside world.” (docs.oracle.com)