Lessons Learned So Far

**Package**

I learnt to structure packages according to their functionality. The modules were decided based on the layers in the software. In case of limited functionality, packages are overkill. The goal is to bundle classes and interfaces such that other programmers can determine the types that are related. This also avoids conflicting of same names as it creates new namespace. Packages types within the package to have unrestricted access to one another yet still restrict access for types outside the package. General convention was packages with utility, model, driver, server and client.

Packages appear to be hierarchical but they are not. So in case only a particular file was used, it makes sense to import that particular file.

I had defined static final fields (constants) and static methods in different packages. Static import gives a way to import constants and static methods without the need to prefix name of the class. Overuse of it can make code difficult to read and maintain since readers wont know of the classes that define the object. But if used with caution, makes code more readable by removing repetition.