Below are key points of what was discussed in the lesson, which was based on the principles outlined in Item 61 of Effective Java book, which says

Item 61: Throw exceptions appropriate to the abstraction

Note: It is ***highly recommended*** to buy the latest edition of Effective Java book by Joshua Bloch. It is one of the must-have books for any serious Java programmer!

Key Points:

- In a multi-layered system, it is not appropriate for a particular higher layer to propagate a lower layer exception when that lower layer exception does not make sense to the task the higher layer is performing. It also confuses the reader when this happens in a really higher-level layer. So, we need to do **exception translation** for that. To allow programmer to investigate lower layer exception, it can be encapsulated within the higher layer exception. This would allow the programmer to extract the lower layer exception programmatically and moreover it would also be part of the higher layer exception's stack trace.

- As much as possible a higher layer exception should try to avoid exception translation. One thing that can be done is to log the lower layer exception using some logging facility