**## Advanced Java Study Guide**

You should be able to explain and apply the following topics:

|  |  |
| --- | --- |
| **### Threads and Garbage Collection** | |
| - Lifecycle of a thread |  |
| - Thread states |  |
| - Creating threads |  |
| - Thread class & thread methods |  |
| - Runnable interface |  |
| - Deadlock and ways to prevent it |  |
| - Livelock |  |
| - How to invoke Garbage Collection |  |
|  |  |
| **### Reflections API** | |
| - Benefits and purpose of Reflection API |  |
| - Classes and interfaces |  |
| - Class |  |
| - Method |  |
| - Modifier |  |
| - Parameter |  |
| - Constructor |  |
|  |  |
| **### Java 8** | |
| - Functional interfaces |  |
| - Lamdba functions |  |
|  |  |
| **### JUnit** | |
| - Test-driven development (TDD) |  |
| - JUnit annotations |  |
| - @Test | declares a method as a test method |
| - @Before | declares a setup method that runs before each test method is run |
| - @After | declares a ‘tear-down’ method that runs after each test method |
| - @BeforeClass | declares a setup method that runs once, before all other methods in the class |
| - @AfterClass | declares a ‘tear-down’ method that runs once, after all other methods in the class |
| - @Ignore | declares that the proceeding test will not be run.  Used to check other test cases within a test class before refactoring individual test cases. |
| - Order of execution of these annotated methods |  |
| - Assert class methods |  |
| - Testing best practices |  |
| - Dependency injection |  |
| - Writing testable code |  |
| - Mocking | Mock is an Object that clone the behavior of a real object. It is basically used in Unit Testing by testing the isolated unit even when Backend is not available. |
| - Measuring code coverage |  |
| - Externalize data |  |
|  |  |
| **### Log4j** | |
| - Benefits / purpose of logging | * Recording unusual circumstances or errors that may be happening in the program * Getting the info about whats going in the application |
| - Log4j configuration |  |
| - Log4j logging levels and threshold |  |
| - ALL | All 8 levels |
| - Trace | finer-grained informational events than DEBUG |
| - DEBUG | designates informational events that are most useful to debug an application |
| - INFO | informational messages that highlight the progress of the application at the coarse-grained level |
| - WARN | designates potentially harmful situations |
| - ERROR | designates error events that might still allow the application to continue running |
| - FATAL | severe error events that presumably lead the application to abort |
| - OFF | highest possible level, intended to turn off logging |