virtusa®

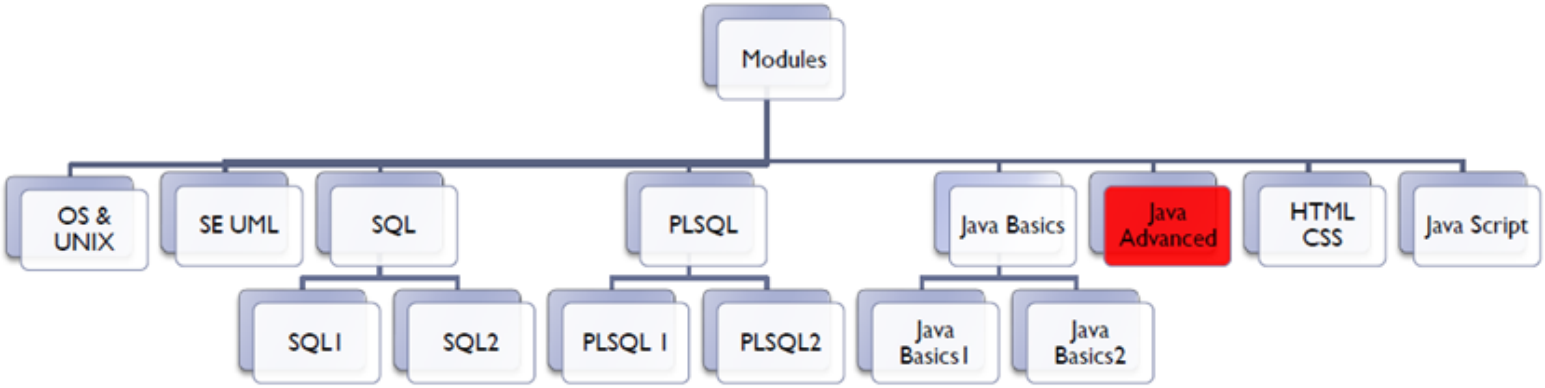
Accelerating Business Outcomes

Java Advanced

Module Overview

Purpose:

- The following module hierarchy presents the technical modules required to build the basic IT skills and acquaints you with relevant technology basics.
- The current module – Java Advanced (highlight in red) underwrites Basics of Java Database connectivity, unit testing, logging using Log4J and will enable you to enhance one's coding skills in JDBC, unit testing and logging.



*Recommended duration to complete Java Advanced module: 12 hours

Module Objectives

By the end of this module, you will be able to:

- Develop persistent Java applications using JDBC API
- Develop test cases using jUnit
- Enable logging for a Java application using Log4J API

Java Database Connectivity

What is JDBC?

JDBC (Java Database Connectivity), is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

Usage of JDBC

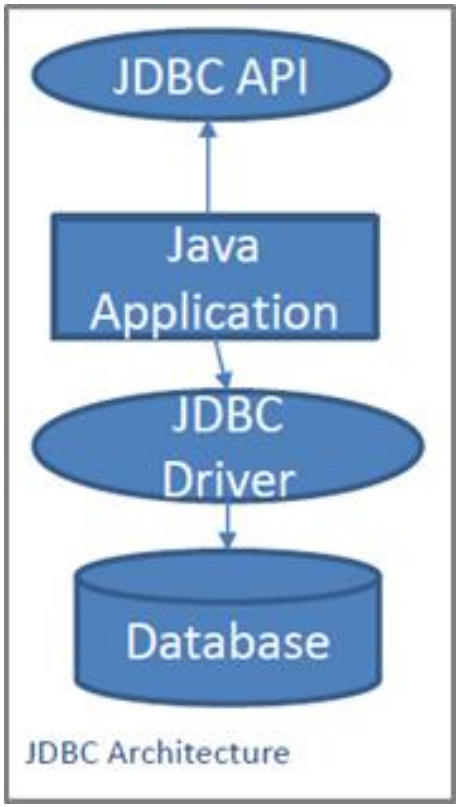
- Making a connection to a database
- Creating SQL or MySQL statements
- Executing that SQL or MySQL queries in the database
- Viewing and Modifying the resulting records

JDBC Architecture:

The JDBC API supports both two-tier and three-tier processing models for database access but in general JDBC Architecture consists of two layers:

JDBC API: This provides the application-to-JDBC Manager connection.

JDBC Driver API: This supports the JDBC Manager-to-Driver Connection.



JDBC Architecture

Reference

- http://www.javatpoint.com/java-jdbc

Unit Testing using jUnit

What is jUnit?

- It is an *open-source testing frame work* for java programmers. The java programmer can create test cases and test his/her own code.
- It is one of the unit testing framework. Current version is ***junit4***.
- To perform unit testing, we need to create test cases.
- The **unit test case** is a code which ensures that the program logic works as expected.
- The ***org.junit*** package contains many interfaces and classes for junit testing such as Assert, Test, Before, After etc.

jUnit in Java

Usage of Unit Testing:

1) Manual Testing: If you execute the test cases manually without any tool support, it is known as *manual testing*. It is time consuming and less reliable.

2) Automated Testing: If you execute the test cases by tool support, it is known as *automated testing*. It is fast and more reliable.

Reference

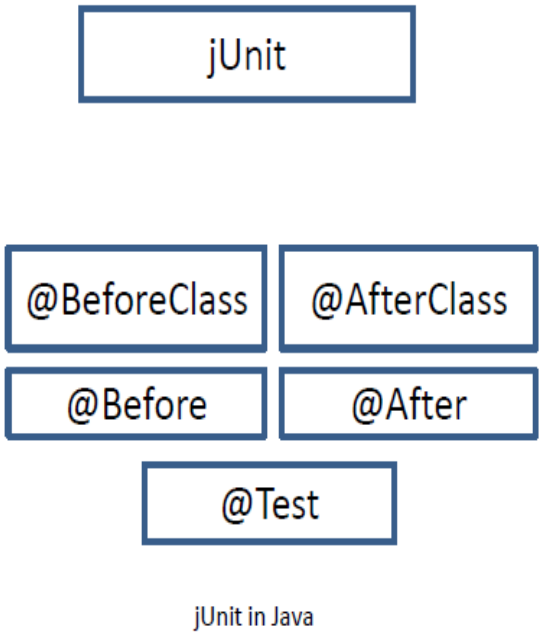
- http://www.vogella.com/tutorials/JUnit/article.html

jUnit Annotations

What is jUnit4 Annotation?

The Junit 4.x framework is annotation based, i.e. the *annotations* that can be used while writing the test cases.

- **@Test** annotation specifies that method is the test method.
- **@Test(timeout=1000)** annotation specifies that method will be failed if it takes longer than 1000 milliseconds (1 second).
- **@BeforeClass** annotation specifies that method will be invoked only once, before starting all the tests.
- **@Before** annotation specifies that method will be invoked before each test.
- **@After** annotation specifies that method will be invoked after each test.
- **@AfterClass** annotation specifies that method will be invoked only once, after finishing all the tests.



jUnit in Java

Assert class: The ***org.junit.Assert*** class provides methods to assert the program logic.

Required jar files: You need to load **junit4.jar** and **hamcrest-core.jar** files.

Reference

- http://www.vogella.com/tutorials/JUnit/article.html

Logging Using Log4J

What is Log4J?

The Junit 4.x framework is annotation based, i.e. the *annotations* that can Log4j is a Reliable, Fast and Flexible *Logging Framework* (APIs) written in Java which is distributed under the Apache Software License.

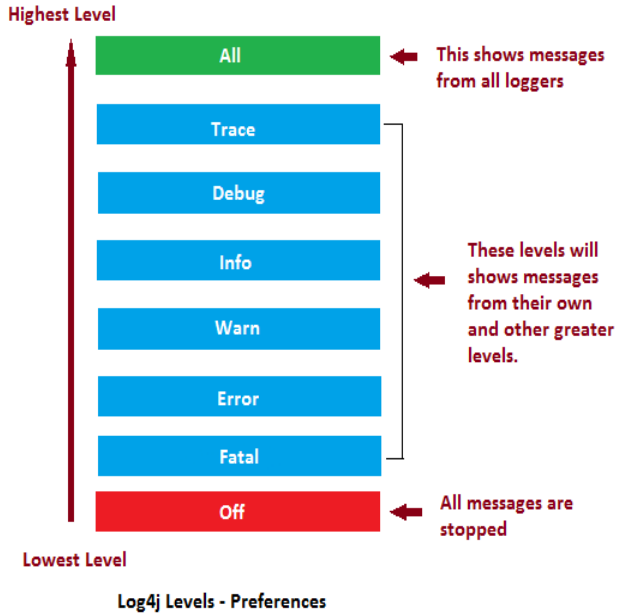
Log4j has been ported to the C, C++, C#, Perl, Python, Ruby, and Eiffel languages.

Usage of Log4J

- Log4j is highly configurable through external configuration files at runtime. It views the logging process in terms of levels of priorities and offers mechanisms to direct logging information to a great variety of destinations, such as a database, file, console, UNIX Syslog etc.

Log4j has three main components:

1. **loggers:** Responsible for capturing logging information.
2. **appenders:** Responsible for publishing logging information to various preferred destinations.
3. **layouts:** Responsible to format logging information in different styles.



Log4j Levels - Preferences

Reference

- http://www.tutorialspoint.com/log4j/

Additional References

To explore more on the subject, refer the below links and books:

Links:

- http://docs.oracle.com/javase/tutorial/jdbc/basics/
- http://www.jdbc-tutorial.com/
- http://www.tutorialspoint.com/jdbc/

Books:

- Java Complete Reference
- Head First Java

Self Check

Instructions to write Self Evaluation Sheet:

Open the excel sheet, refer Java Advanced sheet, write down the solutions for all questions, save a local copy in your machine.

Lab Assignment

- Refer ***Assignment Document*** for this module to proceed with **Lab Assignment.**
- Do **submit the Solutions** for the given assignment and refer the ***Participant guide*** for submission procedure.

Module Summary

Now that you have completed this module, you will be able to:

- Explain -Java and Database connectivity
- Write a persistent Java code with database operations
- Write unit test cases using jUnit
- Configure Logging using Log4J

Thank you