

ITMD 415/515

Advanced Software Development

Week 1 – Introduction and Setup

Instructor: Scott Spyrison

About Class

- ITMD 415
- ITMD 515
- Practical Jakarta EE Enterprise Programming

About Me

- LinkedIn
 - www.linkedin.com/in/scott-spyrison-867b81

About You

- Coursework
 - ITMD 411 or 510 - Intermediate Software Development/OOP (**Pre-req**)
 - ITMD 361 – Web Development
 - ITMD 421 - Data Modeling and Applications
- Helpful Experience
 - Command Line
 - IDE
 - Version Control
 - SQL
 - Web Scripting Languages (JSP, PHP, ASP, etc)

Syllabus Review

Course Tools – On Your Computer

- On Your Computer
 - OpenJDK
 - IDE
 - IntelliJ IDEA
 - **Apache NetBeans**
 - Eclipse
 - Whatever – as long as it supports maven projects
 - Application Server
 - Payara
 - Database
 - MySQL

Course Tools – Online

- In The Cloud
 - Canvas or Beacon LMS
 - YouTube
 - GitHub

Canvas/Beacon vs Other Tools

- Canvas or Beacon LMS
 - Syllabus
 - Announcements
 - Assignments and Submissions
 - Gradebook, Grades and Feedback
 - Your Questions and Answers
 - My Questions and Answers
- Github
 - Your Lab src and Documentation (README)

Academic Honesty

- Your work must be your own
- Collaboration vs Cheating
 - Collaboration is a valued skill
 - Cheating is unprofessional and unethical
- I strongly recommend you do not cheat
- If you do, I *will* find out and I will be required to take action

What is Enterprise?

- Enterprise vs Startup
- Motivational Thoughts and Market Relevance

What Does an Enterprise Need?

- Standards
 - Specification
 - Implementation
 - Reference Implementation
- Process
- Quality
- Consistency and Reproducibility
- Flexibility and Scalability
- Productivity

The Road to Jakarta EE

- <https://blogs.oracle.com/theaquarium/opening-up-ee-update>
- <https://blogs.oracle.com/theaquarium/the-road-to-jakarta-ee>
- <https://javaee-guardians.io/2018/03/02/java-ee-is-officially-retired-its-now-called-jakarta-ee/>

Where are we Today?

- <https://jakarta.ee/>
- <https://projects.eclipse.org/projects/ee4j>
- <https://projects.eclipse.org/proposals/eclipse-glassfish>
- <https://netbeans.apache.org/>

Jakarta EE

As defined by Antonio Goncalves in *Beginning Java EE 7*:

“Java EE is a set of *specifications* implemented by different *containers*. Containers are Java EE runtime environments that provide certain *services* to the *components* they host...”

Java SE vs Jakarta EE

- Jakarta EE is **in addition** to Java SE, not **instead of**
- It all begins with the Java programming language and JVM
- Think of Jakarta EE as an extension of Java SE
- Think of Jakarta EE as the back-end or server
- Java SE – Run a program
- Jakarta EE – Run an application on a server
 - Deploy an application to an application server
- Jakarta EE – Run a micro-service
 - Application server embedded in application
 - Runs as a self-contained service

Containers/Application Servers

- Container? What, Docker? Huh?
- Glassfish
- Wildfly
- Open Liberty
- **Payara**
- JBoss
- Weblogic
- Websphere
- Tomcat

<http://www.oracle.com/technetwork/java/javaee/overview/compatibility-jsp-136984.html>

Past EE Criticisms

- <https://www.youtube.com/watch?v=47730LFtemY>
- Complexity, Size, Bloat. Y2K not pretty for J2EE (*as it was called at the time*)
- Spring came into being ~ 2002/2003
- But then eventually came...
 - Maven (and others) for dependencies
 - Annotations
 - EE 6/7/8 and Profiles
 - Simplified Packaging
 - Bliss and happiness

Jakarta EE Programming Model

Simplified

Pojo + Metadata

Pojo = Plain Old Java Object.

Metadata = Annotations or XML

Remember Annotations from 411? Annotations are an example of declarative programming. They are not “part of the program” but provide data about the program to the compiler or run-time.

Dependency Injection

Cat myCat = ***new*** Cat() vs ***@Inject*** Cat cat

Jakarta EE Programming Model

Java SE annotation examples:

@Override

@SuppressWarnings

@Documented

Jakarta EE annotation examples:

@Entity

@Servlet

@ManagedBean

@WebService

Standards Revisited

- A little deeper into Standards
 - Specification
 - Implementation
 - Reference Implementation
- Specification : Interface
 - Can you compile your code against an interface?
 - Can you run it against an interface?
- Interface : Contract
- Implementation : Runtime
- Reference Implementation : Gold Standard

Underlying Strategies

- Separation of Concerns
- Containers (provide services - super-JVMs)
- Configuration by Exception/Convention over Configuration
- Declarative Programming
- Developer Tools and *Scaffolding* (Productivity)
- Inversion of Control (<http://bit.ly/1pCY23A>)
- Dependency Injection
- Cross-Cutting/Aspect Oriented

Lab 1 - Setup and Demonstration

Platform Basics – NB and Payara

- IDE
 - Overview
 - Projects, Files and Services
 - Basic NetBeans Webapp
- Payara
 - Ports 8080/4848
 - Starting/Stopping
 - Deployment/Undeployment
 - Log Files

Platform Basics – Git and Maven

- Git
 - D(istributed)VCS
 - Operations
 - Clone
 - Add and Commit
 - Push
- Maven
 - POM and Project Structure
 - Dependencies
 - Lifecycle

Homework

- Lab 1 - Setup Tools and Environment
- Ask Questions in the Discussion Forum
- Answer Questions in the Discussion Forum

Sources Used

- The Jakarta EE Tutorial. Retrieved Aug 24, 2020, from <https://eclipse-ee4j.github.io/jakartaee-tutorial/toc.html>
- Juneau, J. (2020). Jakarta EE 8 Recipes. New York, NY: Apress.
- Goncalves, A. (2013). Beginning Java EE 7. New York, NY: Apress.
- Some slides adapted with permission from Marty Hall (www.coreservlets.com – JSP and Servlets)