

## 1. W1D1

- a. In the morning, we learned / discussed about:
  - i. Business development leads to application development: all code (data access, business, render code...) in a single file -> 3 tiers MVC -> N tiers.
  - ii. When an application grows larger (monolith), it can be splitted / broken to (micro) services (service oriented) to mitigate maintainability issues such as a change in one place could affect many other things or recompile the whole application, multiple teams work on a big single application is more difficult. Of course, integration is needed for services. There are some approaches to integrate / communicate between services such as remote invocation and messaging.
  - iii. Concepts, Principles and patterns: Domain Driven Design (DDD), IoC, DI (loose coupling), Aspect Oriented Programming (AOP), Interceptors, Object Relational Mapping (ORM), Repositories aka Data Access Objects (DAO), Data Transfer Objects (DTO).
  - iv. Java Spring: 4 ways to connect objects together, Spring components, and how Spring works.
- b. In the afternoon, we practiced how to setup, configure and work with Java, Maven, MySQL and Tomcat.

## 2. W1D2

- a. In the morning, we learned / discussed about:
  - i. Dependency Injection (DI): what it is, how it works, features, how to use/configure it in the different ways, when/what to use with advantages of each way.
  - ii. Spring Startup Order, Call Order, Life Cycle Methods, Lazy and Eager
  - iii. Spring Creation Order
- b. In the afternoon, we practiced how to implement / configure DI.

## 3. W1D3

- a. In the morning, we learned / discussed about:
  - i. Aspect Oriented Programming (AOP): what it is, Cross Cutting Concerns, Interceptor pattern.
  - ii. AOP Terminology: Advice, JoinPoint, Pointcut, Aspect, ...
  - iii. AOP Advices: 5 types
  - iv. Pointcut Expression Language
  - v. Proxy Weaving
- b. In the afternoon, we practiced AOP and DI.

## 4. W1D4

- a. In the morning, we learned / discussed about:
  - i. Java Persistence API
  - ii. Hibernate: Entity Mapping, Identity Mapping, ...
  - iii. Lombok
- b. In the afternoon, we practiced Hibernate.

## 5. SCI

- a. The whole is greater than the sum of the parts, since it matters just as much how the parts are connected. For example, many services connected together to compose the whole system.
- b. Unity in Diversity. You can get the same result in many ways. For example, there are many ways to inject the same dependency.