The Learning House

427 S. 4th Street #300

Louisville KY 40202

Spring Core Unit –
Software Development Lifecycle

Lesson 3 - Spring IOC Container

SOFTWARE GUILD

Objectives

- Learn how to create a Spring application context XML file
- Learn how to declare beans
- Learn how to inject beans using both constructors and setters
- Understand alternative configuration options

Let's Build Something...

- Starting with the Spring DI container
- We'll start with a couple of simple console applications
- All skills covered will transfer to our MVC web applications

Example: Olympian

- New Maven project
- Swap in JUnit 4.11
- Implement w/o DI
- Implement w/DI

DI with Spring

- Set up the application context
  - Get skeleton file from GitHub
- We now have to add dependency entries to our Maven POM file
- applicationContext.xml
  - I have added all the namespace entries that we'll encounter — use as a template for all projects

Create Classes and Define Beans

- Create classes, like always
- Define beans in applicationContext.xml
- We can now create the context in our Java code and then ask it for our objects
  - Note: we don't create these objects, Spring does
  - Ex: SkiJumper — just a normal class, no DI

Constructor Injection

- Create and define Olympian class
- Create and define Event class
- Ask for Olympian from context
- What can go wrong?
  - Misspelled bean name
  - Malformed XML
  - Use ‘value’ when we need to use ‘ref’

Property Injection

- Define another bean of type Olympian
- Make the Olympian be from Canada

What About Multi-Event Athletes?

- We can inject Collections
- Create MultiSportAthlete

Advanced Topics

- SpeL — can use scripting in your XML files for conditional wiring
- Left as an exercise for the reader
- **Use sparingly** — overuse results in a lot of application logic programmed in XML, which can be very ugly and hard to debug

Advanced Topics (2)

- I prefer XML Wiring, but there are other options:
  - Autowiring and Inject annotations
    - We will use this for some wiring in MVC
  - Writing configurations in Java