Training: Capstone Courseware
117B Java Persistence with Spring

TRAINING GOALS:

This course enables the experienced **Java developer** to use the **Spring application framework** to manage objects in a lightweight, inversion-of-control container, and to build persistence components and application tiers using Spring's support for relational databases and transaction control.

Spring's core module gives the developer declarative control over object creation and assembly; this is useful for any tier of any Java application, so we study it in some depth to begin the course. Then students build persistence code using the Spring DAO and ORM modules, using both JDBC and JPA, with Hibernate as the provider. We study both programmatic and declarative transaction control. The course concludes with a chapter on Spring's testing framework, with a focus on test-managed transactions.

Learning Objectives

- Understand the scope, purpose, and architecture of Spring
- Use Spring application contexts to declare application components, rather than hard-coding their states and lifecycles
- Use dependency injection to further control object relationships from outside the Java code base
- Use annotations to take advantage of Spring post-processors for automated bean instantiation and wiring
- Connect business objects to persistent stores using Spring's DAO and ORM modules
- Simplify JDBC code using Spring templates
- Integrate JPA entities and DAOs into Spring applications
- Control transactions using Spring, either programmatically or declaratively
- Develop effective unit tests using Spring's test framework

CONSPECT:

- Overview of Spring
  - Java EE: The Good, The Bad, and the Ugly
  - Enter the Framework
  - Spring Value Proposition
  - The Spring Container

Compendium Education Center Ltd.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Bielska 17, 02-394 Warszawa, tel.: (22) 417 41 70

BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- Web Applications
- Persistence Support
- Aspect-Oriented Programming
- The Java EE Module(s)

- The Container
  - JavaBeans, Reconsidered
  - The Factory Pattern
  - Inversion of Control
  - XML View: Declaring Beans
  - Java View: Using Beans
  - Singletons and Prototypes
- Instantiation and Configuration
  - Configuring Through Properties
  - Configuration Namespaces
  - The p: Notation
  - Bean (Configuration) Inheritance
  - Configuring Through Constructors
  - Bean Post-Processors
  - Lifecycle Hooks
  - Integrating Existing Factory Code
  - Awareness Interfaces
- Dependency Injection
  - Assembling Object Graphs
  - Dependency Injection
  - Single and Multiple Relationships
  - The Utility Schema
  - Using Spring Expression Language (SpEL)
  - Inner Beans
  - Autowiring
  - @Component, @Service, & Company
  - @Autowired Properties
  - Best Practices with Spring Annotations
  - Java Classes as @Configurations
  - AnnotationConfigApplicationContext
  - Capabilities and Limitations
  - Mixing and Importing XML and Java Configurations

Compendium Education Center Ltd.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Bielska 17, 02-394 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- Assembling Object Models
  - Collections and Maps
  - Support for Generics
  - The Spring Utility Schema (util:)
  - Autowiring to Multiple Beans
  - Order of Instantiation
  - Bean Factory vs. Application Context
- Persistence with JDBC
  - Reducing Code Complexity
  - The DataAccessException Hierarchy
  - JdbcTemplate
  - RowMapperand ResultSetExtractor
  - The DaoSupport Hierarchy
  - Capturing Generated Keys
  - Transaction Control
  - PlatformTransactionManager
  - TransactionTemplate
  - Isolation Levels
  - Transaction Propagation
- Persistence with JPA
  - Object/Relational Mapping
  - The Java Persistence API
  - JpaDaoSupport and JpaTemplate
  - @PersistenceUnit and @PersistenceContext
  - Shared Entity Managers
  - Using
  - The @Transactional Annotation
  - Isolation and Propagation
  - A Limitation of @Transactional
  - Understanding Entity States
  - Configuring JPA Without persistence.xml
  - Bean Validation in JPA
  - Optimistic Locking
- Testing
  - Testability of Spring Applications
  - Dependency Injection

Compendium Education Center Ltd.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Bielska 17, 02-394 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007

- Mocking
- SpringJUnit4ClassRunner
- TestContext
- @ContextConfiguration
- Preserving Test Isolation
- @DirtiesContext
- AbstractTransactionalJUnit4SpringContextTests
- Testing JPA Components
- Mixing JPA and JDBC
- TestTransaction
- Profiles

REQUIREMENTS:

- **Java programming** is required -- Course 103 is excellent preparation.
- Basic knowledge of XML is recommended -- Course 501 **Introduction to XML**.
- For the final chapter some undersatnding of **JUnit** is required -- Course 191 **Introduction to Java Testing**.

Difficulty level

CERTIFICATE:

The participants will obtain certificates signed by Capstone Courseware.

TRAINER:

Authorized Capstone Courseware Trainer.

Compendium Education Center Ltd.
ul. Tatarska 5, 30-103 Kraków, tel.: (12) 298 47 77
ul. Bielska 17, 02-394 Warszawa, tel.: (22) 417 41 70
BNP Paribas Bank Polska S.A.
EUR: PL 79 1600 1462 1853 6451 3000 0006
USD: PL 52 1600 1462 1853 6451 3000 0007