Training Syllabus for LEAP 2023 FSE Track (US and IRE)

ROITRAINING
MAXIMIZE YOUR TRAINING INVESTMENT

Fidelity LEAP
Technology Immersion Program

This page intentionally left blank.

Table of Contents

This page intentionally left blank.

Training Syllabus for LEAP 2023 FSE Track

This outline provides an ROI training 'roadmap' of the expected activities for each day. The curriculum consists of lectures, hands-on code exercises, classroom discussion, and group assignments aligned to facilitate the development of skills utilized by Full Stack Engineers on the job. There are four skills assessments and one knowledge check included within the curriculum:

Type	Name
Assessment	Developing Client-Side Dynamic Web Applications
Assessment	Programming with Java
Assessment	Working with Relational Databases
Knowledge Checkpoint	Mastering Spring with MyBatis
Assessment	Developing RESTful Services w/ Java + Spring

Fidelity will supplement the ROI virtual instructor-led training (VILT) courses using on-line training (Pluralsight) as well as hands-on activities utilizing a simulated project. A Technical Coach is assigned to each classroom to facilitate the end-to-end development of team project solutions and manage the transition between online training to instructor-led.

Course Materials:

Each associate receives the following:

- Course notes
- Exercise materials
- Recommended reading and associate additional resources

Evaluation:

Progress of associates is determined by the ability to:

- Understand and apply assigned development requirements
- Read directions, pay close attention to details, and produce correct code
- Finish assignments within the allotted time
- Participate in team project activities
- Demonstrate understanding of curriculum via knowledge and skill assessments

The following course outlines are inclusive of the entire ROI Full Stack Engineering track courses. The daily schedule (link below) outlines specific ROI course chapters to be covered during ROI instructor-led training.

Attachment A: FSE Track – R1 ROI Course Schedule

Developing Client-Side Dynamic Web Applications

Learning Objectives:

- Learn to develop responsive web applications
- Understand what makes a web application responsive and interactive
- Build interactive websites with HTML
- Employ behavior in web pages using JavaScript and AJAX
- Standardize presentation using Cascading Style Sheets (CSS)
- Leverage the capabilities of HTML5 and CSS3
- Understand the need for JavaScript frameworks such as jQuery, node.js, and Angular
- Perform End-to-End (E2E) testing with Jasmine and Protractor

Technologies Covered:

- HTML
- CSS
- JavaScript
- jQuery
- Jasmine Testing Framework
- Angular CLI
- TypeScript
- Karma Test Runner
- Cypress E2E Testing Framework

Course	Chapter	Overview	Skills Covered
Dynamic Web Applications	1	**As a Full Stack Engineer,** understand the basics of Visual Studio Code to launch web pages using the software.	**Introduction to Visual Studio Code** • Explorer • Extensions • Work area • Launching web page from Visual Studio Code
Dynamic Web Applications	2	**As a Full Stack Engineer,** learn how HTML and CSS Styles can be used to standardize presentations.	**Standardizing Presentation with HTML and CSS** • HTML Essentials • Styling with CSS
Dynamic Web Applications	3	**As a Full Stack Engineer,** explore further features of HTML and CSS to create more advanced web pages.	**Advanced HTML and CSS** • HTML Layout • The Document Object Model • iFrame • Tables • Advanced CSS Effects • Semantic Tags • Forms and Validation
Dynamic Web Applications	4	**As a Full Stack Engineer,** apply JavaScript to add dynamic behavior to a web page.	**Client-Side JavaScript Programming** • JavaScript Syntax • Working with the DOM • Handling Events • Web Application Security Client Side
Dynamic Web Applications	5	**As a Full Stack Engineer,** learn how to use JavaScript jQuery framework to simplify building interactive web pages.	**Working with jQuery** • Getting Started with jQuery • jQuery Selectors • Event Methods • Filtering • Ajax with jQuery

Course	Chapter	Overview	Skills Covered
Dynamic Web Applications	6	**As a Full Stack Engineer,** get started writing reactive Single Page Applications using Angular by learning to use TypeScript and understanding how to incorporate automated testing into the development process.	**Introduction to Angular** • Angular • Introducing TypeScript • A Simple Angular Application • Angular Test Utilities
Dynamic Web Applications	7	**As a Full Stack Engineer,** learn how to create an Angular component, define strongly typed classes and pass data between a parent and child component to apply these skills when building Angular applications.	**Angular Components** • Creating Angular Components • Built-In Directives • An Angular Application
Dynamic Web Applications	8	**As a Full Stack Engineer,** learn how to create an Angular module; bind properties, attributes, and events; use two-way binding with HTML inputs to apply these skills when building Angular applications on the job.	**Angular Modules and Binding** • Organizing Code into Modules • Binding Data, Methods, and Events
Dynamic Web Applications	9	**As a Full Stack Engineer,** understand the role of services, build a service to retrieve data, make GET and POST calls using the HttpClient service, and process data returned from the HttpClient service.	**Angular Services** • Dependency Injection • Creating Services • RESTful Services • Using the HttpClient Service • Handling Errors
Dynamic Web Applications	10	**As a Full Stack Engineer,** understand the importance of End-to-End testing (E2E) and learn to implement E2E tests for Angular applications to apply these skills and perform E2E tests on the job.	**Angular End-to-End (E2E) Testing Applications** • End-to-End Testing • Introducing Cypress Advanced Cypress
Dynamic Web Applications	11	**As a Full Stack Engineer,** apply skills learned throughout the course so far to build a fully functional Angular application.	**Building an Angular Application**
Dynamic Web Applications	12	**As a Full Stack Engineer,** learn the use of and how to build a pipe to transform and display useful data inside a template.	**Pipes** • Introducing Pipes • Building Custom Pipes
Dynamic Web Applications	13	**As a Full Stack Engineer,** understand how routing is used to build a Single Page Application.	**Angular Routing** • Routing in SPAs • Introducing the Angular Router • Parameterized Routes • Advanced Routing

Dynamic Web Applications	14	**As a Full Stack Engineer,** learn how to build and validate both template- and model-driven forms to interact with data in an Angular application.	**Angular Forms** • Introducing Angular Forms • Template-Driven Forms • Model-Driven Forms
Dynamic Web Applications	15	**As a Full Stack Engineer,** learn how to use the Angular Environment to build and deploy applications to apply these same skills when building Angular applications on the job.	**Angular Deployment** • Angular Environment • Building Angular Applications Deploying Angular Applications

Course Appendices

Appendix A: Angular Directives

Appendix B: Observables

Programming with Java

Learning Objectives:

- Practice TDD in Java
- Write Java code to represent UML classes
- Apply Inheritance in Java
- Use effective programming practices
- Use a variety of classes to support different goals
- Use abstract classes and interfaces
- Refactor code to improve design while developing code
- Implement exceptions to handle unusual issues
- Effectively utilize the functional programming features of Java

Technologies Covered:

- Java
- Object-Oriented Programming (Polymorphism, Inheritance, Abstraction, Encapsulation)
- Refactoring
- JUnit Testing Framework
- Exception Handling
- Functional Programming in Java

Course	Chapter	Overview	Skills Covered
Programming with Java	1	**As a Full Stack Engineer,** learn to use Java classes and explore the existing String class to create instances of objects.	**Introduction to Java** • Basic Java Syntax • Creating and Calling Objects • Using Existing Classes: String
Programming with Java	2	**As a Full Stack Engineer,** learn how to use JUnit and associated concepts to effectively determine methods and values for test inclusion and effectively write and test code.	**Test-Driven Development** • TDD with JUnit • Writing Tests • Test Fixtures • More JUnit • Debugging
Programming with Java	3	**As a Full Stack Engineer,** learn how to program defensively to ensure code appropriately implements design; overload constructors and methods judiciously; and think before implementing getters and setters to successfully program applications in an efficient yet effective manner.	**Implementing Classes** • Constructors, Getters, and Setters • Implementing Behavior with Methods
Programming with Java	4	**As a Full Stack Engineer,** learn how Inheritance can lead to reusable code and eliminate duplicate code to produce well-implemented designs.	**Implementing Polymorphism** • Inheritance • Guidelines
Programming with Java	5	**As a Full Stack Engineer,** learn how and when to override the standard methods of the superclass Object when writing code and programming applications.	**Effective Programming** • Object's Standard Methods • equals • hashCode • toString • Methods to Treat Cautiously
Programming with Java	6	**As a Full Stack Engineer,** learn how to apply common and useful Java classes to make Java programming more efficient.	**Useful Classes** • BigDecimal • Date and Time Classes
Programming with Java	7	**As a Full Stack Engineer,** learn how to use constants and enumerations when writing code to provide solutions that meet stated requirements.	**Constants and Enumerations** • Constants • Enumerated Types • Static Fields and Methods

Course	Chapter	Overview	Skills Covered
Programming with Java	8	**As a Full Stack Engineer,** learn how to decide whether to use an interface or an abstract class when writing code to leverage and produce highly extensible code.	**Abstract Classes and Interfaces** • Abstract Classes • Interfaces • Serializable
Programming with Java	9	**As a Full Stack Engineer,** understand how to choose the right collection type based on problem being solved to successfully implement code.	**Java Collections Framework** • Collections Framework • ArrayList • Using Collections • Primitive Wrapper Classes • Sorting
Programming with Java	10	**As a Full Stack Engineer,** learn to use refactoring when programming to improve code design and validate that no bugs were introduced.	**Refactoring** • Duplicate Code • Recognizing Polymorphism • Dependency Inversion Principle
Programming with Java	11	**As a Full Stack Engineer,** learn to throw and test for exceptions to identify and handle errors separately, outside the normal program flow.	**Exceptions** • Exceptions • Testing Exceptions • Custom Exceptions • Using Exceptions Effectively
Programming with Java	12	**As a Full Stack Engineer,** learn to use layers and packages to organize your programs to apply good design practices at all levels of your program.	**API Design** • Packages • Application Layers • API Design • Service Objects
Programming with Java	13	**As a Full Stack Engineer,** learn to apply design patterns to provide proven solutions to commonly occurring software problems.	**Implementing Design Patterns** • Using Design Patterns • Examples of Design Patterns in Java • Factories: Creating Objects • Refactoring to Design Patterns

Appendix A: A Java Basic Syntax

Appendix B: Gang of Four Design Patterns

Working with Relational Databases

Learning Objectives:

- Examine the role of SQL and the basic toolset
- Select, filter, and sort data from the database
- Manipulate the data with Oracle functions
- Extract data from multiple tables
- Aggregate data using group functions
- Create and manage tables, views, and indexes
- Program with PL/SQL
- Create stored procedures, functions, and packages
- Create and work with triggers
- Understand data quality and data movement

Technologies Covered:

- SQL
- Data Manipulation Language (DML)
- Data Definition Language (DDL)
- Transaction Control Language (TCL)
- JDBC
- Testing of Database Transactions in Java
- UT-PL/SQL Unit – Testing Framework
- Big Data Concepts
- BASE and ACID
- NoSQL Data Models
- Amazon DynamoDB

Course	Chapter	Overview	Skills Covered
Relational Databases	1	**As a Full Stack Engineer,** explore core SQL concepts, data modeling, and SQL Developer to use these skills during development and testing.	**What Is Structured Query Language?** • SQL Overview • Designing a Database • The Course Environment • Using SQL Developer
Relational Databases	2	**As a Full Stack Engineer,** learn to build basic SELECT statements using common clauses to apply such skills when building business applications.	**SQL Query Syntax** • Building Basic SELECT Statements • The WHERE Clause • The ORDER BY Clause
Relational Databases	3	**As a Full Stack Engineer,** learn about common data types and functions to use them when building business applications.	**SQL Scalar Functions** • Basic Server Datatypes • Introduction to Functions • Scalar Functions
Relational Databases	4	**As a Full Stack Engineer,** learn various types of join to use them to meet different business requirements.	**SQL Joins** • The Need for Joins • Inner Joins • Outer Joins • Self Joins • Cartesian Joins and SQL-89 Syntax
Relational Databases	5	**As a Full Stack Engineer,** learn about date- and null-handling functions to use them when building business applications.	**Additional SQL Functions** • DATE Functions • Miscellaneous Functions
Relational Databases	6	**As a Full Stack Engineer,** learn how to manipulate data using DML and Transactional Control statements to use them when building business applications.	**Data Manipulation Language** • INSERT • UPDATE • DELETE • Transactional Control
Relational Databases	7	**As a Full Stack Engineer,** learn how to interact with a relational database from Java and avoid common security pitfalls to write business applications that use data effectively.	**Databases with JDBC (Java Database Connectivity)** • JDBC • Executing Queries • Secure Database Access • Implementing a Data Access Object

Course	Chapter	Overview	Skills Covered
Relational Databases	8	**As a Full Stack Engineer,** learn how to update, as well as query, the database; support stored procedures and manage transactions to write data-driven business applications. Learn how to manipulate data using Data Manipulation Language (DML), execute DML commands with JDBC manage transactions (ACID, Isolation levels), and use transactions in testing.	**Updating Databases** • Insert New Records • Transactions • How to Test JDBC • Delete and Update Records • Work with Multiple Tables
Relational Databases	9	**As a Full Stack Engineer,** explore and practice debugging Data Access Object methods, using a Business Service, and testing a Business Service	**Working with a Data Access Object** • Debug Data Access Objects • Use a Business Service • Test a Business Service
Relational Databases	10	**As a Full Stack Engineer,** learn advanced JDBC techniques to maximize code quality and how to use transactions to simplify testing to build more robust business applications.	**Advanced JDBC** • Handling NULL • BigDecimal • Enumerated Types • Dates and Times • Testing with Transactions
Relational Databases	11	**As a Full Stack Engineer,** learn how to define and aggregate groups of data to apply this frequently used technique to meet business requirements.	**Aggregating Information** • Aggregate Functions • The GROUP BY Clause • The HAVING Clause • JOINing Issues • Subqueries
Relational Databases	12	**As a Full Stack Engineer,** learn how to use Set Operators to write business applications that interact with data efficiently.	**Set Operators** • Set Operators
Relational Databases	13	**As a Full Stack Engineer,** learn the core syntax of PL/SQL, including variables; conditional, iterative, and sequential control statements; error handling; and cursors, to improve the design and structure of business applications that handle relational databases.	**Programming with PL/SQL** • Working with PL/SQL • Control Structures and Exceptions • Cursors
Relational Databases	14	**As a Full Stack Engineer,** learn how to store PL/SQL code in the database to write more efficient data-processing applications.	**Creating Stored Procedures, Functions, and Packages** • Procedures and Functions • Packages • Debugging

Course	Chapter	Overview	Skills Covered
Relational Databases	15	**As a Full Stack Engineer,** learn how to apply unit testing techniques to PL/SQL code to create more robust business applications.	**Testing PL/SQL** • utPLSQL • Testing Updates
Relational Databases	16	**As a Full Stack Engineer,** create and use database triggers to address specific business requirements.	**Creating Triggers** • Statement-Level and Row-Level Triggers • Conditional Predicates • Managing Triggers
Relational Databases	17	**As a Full Stack Engineer,** understand how to create and modify important schema objects to develop a database that meets business requirements.	**Data Definition Language** • Schemas • Create and Manage Tables • Sequence Generators • Enforcing Database Integrity • Managing Views • Managing Indexes • Using SQL Developer
Relational Databases	18	**As a Full Stack Engineer,** explore the factors that impact how a relational database may be used to support cloud-scale data to design databases that deal effectively with the large data volumes and throughput that are often encountered with modern client-facing applications.	**Scaling Relational Databases** • Scaling Relational Databases • Relational Databases in the Cloud
Relational Databases	19	**As a Full Stack Engineer,** understand what Big Data is and how typical NoSQL stores work, to select the right type of store for different business requirements.	**Big Data and NoSQL** • Big Data • Introduction to NoSQL • Types of NoSQL Store • NewSQL
Relational Databases	20	**As a Full Stack Engineer,** explore Amazon DynamoDB; learn how to interact with DynamoDB at its native (low) level; and, understand how the Java Object Mapper allows a more object-oriented interaction to work with managed NoSQL database services that provides fast and predictable performance with seamless scalability.	**Amazon DynamoDB** • DynamoDB • Core Concepts • AWS CLI • Java API • Java Object Mapper • Advanced Topics

Mastering Spring and MyBatis

Learning Objectives:

- Use the Spring framework to build clean, extensible, loosely coupled enterprise Java applications
- Utilize Spring as an object factory and dependency injection to wire components together
- Understand and apply MyBatis to simplify access to relational databases
- Explore and apply Spring to simplify the use of MyBatis in an application
- Apply transaction strategies via configuration

Technologies Covered:

- Spring Framework
- MyBatis
- XML and Annotation-Based Configuration
- JUnit Testing in Spring
- Functional Programming

Course	Chapter	Overview	Skills Covered
Spring and MyBatis	1	**As a Full Stack Engineer,** understand the Spring Framework and use of the general-purpose object factory to simplify the development of effective applications in Java.	**Introducing the Spring Framework** • The Spring Object Factory • Annotation-Based Factory Configuration
Spring and MyBatis	2	**As a Full Stack Engineer,** learn more about Spring dependency injection and how, and when, to use it in tests to create more robust business applications.	**Understanding Spring** • Spring and Dependency Injection • Testing with Spring • Working with Maps • Other Dependency Types
Spring and MyBatis	3	**As a Full Stack Engineer,** learn better ways of defining dependencies including using expressions and multiple Spring configurations; and, how to debug Spring configuration problems, to build better structured and more maintainable business applications.	**Advanced Spring Configuration** • Managing Bean Lifecycle • Expression Language • More Configuration Options • Debugging Spring Configuration Problems
Spring and MyBatis	4	**As a Full Stack Engineer,** learn why Object-Relational Mappers are needed; how to configure and invoke MyBatis from Spring; and, how to define complex data relationships in MyBatis, to leverage the framework to create business applications that can interact more effectively with relational data.	**Introduction to MyBatis and Spring** • Configuring a Data Source • Domain Store Design Pattern • Configuring MyBatis with Spring • Querying a Database with MyBatis in Spring • Working with Relationships
Spring and MyBatis	5	**As a Full Stack Engineer,** perform database update operations with MyBatis; learn how Spring supports using transactions in tests; and, how to configure MyBatis with annotations, to use the framework more effectively in robust business applications.	**Working Effectively with MyBatis** • DML Through MyBatis with XML • Transaction Management in Testing • Advanced Topics • SQL Mappers Using Annotations
Spring and MyBatis	6	**As a Full Stack Engineer,** learn how to use functional programming techniques in Java to create business applications that are more maintainable.	**Functional Programming** • Functional Programming • Lambda Expressions • Stream API • Functional Interfaces • Optional Variables

Developing RESTful Services

Learning Objectives:

- Solve common programming problems by using design patterns
- Design and build RESTful web services
- Use JAX-RS and Spring Boot to create RESTful web services written in Java
- Use Node.js to execute RESTful services written in JavaScript
- Use some advanced JavaScript programming techniques

Technologies Covered:

- RESTful Services in Java
- Testing RESTful Services
- Spring Support for RESTful Services
- Security Techniques for RESTful Services
- Cloud Design Pattern
- Node.js
- Express Framework
- RESTful Services with JavaScript and Node.js
- Jasmine Testing Framework
- Cucumber Testing Framework
- Higher Order Functions and Other Advance JavaScript Techniques

Course	Chapter	Overview	Contents
RESTful Services	1	**As a Full Stack Engineer,** learn the benefits of building REST-based web services with Spring Boot to simplify communication protocols. Learn and use HTTP response codes to design effective RESTful APIs.	**Building RESTful Services** • Understanding RESTful web services • What is Spring Boot? • Building RESTful services with Spring Boot • Returning HTTP Status Codes
RESTful Services	2	**As a Full Stack Engineer,** learn how to design an effective and efficient RESTful web service API. Learn to deploy a Spring Boot-based RESTful web service. Research the use of Spring Boot at Fidelity.	**Designing RESTful Services** • Designing an effective RESTful API • RESTful API guidelines • Deploying a Spring Boot based RESTful Web Service • The Twelve Factor App • How Spring Boot is Used at Fidelity • How to Debug a RESTful Web Service
RESTful Services	3	**As a Full Stack Engineer,** learn how to use a variety of techniques to test RESTful web services. Perform end-to-end testing of a RESTful Service.	**Testing RESTful Services** • Discuss the responsibilities of a RESTful Service Controller • Perform unit testing of POJOs in the back end • Test the RESTful Service controller in the web (HTTP) environment • Perform end-to-end testing of a RESTful Service using TestRestTemplate
RESTful Services	4	**As a Full Stack Engineer,** apply techniques for securing RESTful web services; examine session management practices; manage authentication and authorization; and, explore the details of OAuth for authorization to integrate security best practices within development activities.	**Securing RESTful Web Services** • Session Management • Authentication and Authorization • Securing RESTful Web Services
RESTful Services	5	**As a Full Stack Engineer,** explore challenges in developing applications for cloud deployment; and, review patterns and functions to apply to mitigate or minimize encounters.	**Cloud Design Patterns** • Challenges in Cloud Development • Cloud Patterns • AWS Lambda Functions

Course	Chapter	Overview	Contents
RESTful Services	6	**As a Full Stack Engineer,** learn how to use Node.js to become a web server; and, how to access databases from JavaScript to apply RESTful best practices on the job.	**Node.js** • Features and First Steps • The package.json File • Basic Database Access • JSON data • mock File
RESTful Services	7	**As a Full Stack Engineer,** learn how to debug and test a Node.js application and utilize Express to program faster RESTful services with Node.js to apply RESTful best practices on the job.	**Node.js and Express** • Debugging a Node Application • Working with Node and Express • CORS (Cross • Origin Resource and Sharing) • Building RESTful Services Faster with Express • Basic Use of Jasmine with Node.js and Express
RESTful Services	8	**As a Full Stack Engineer,** work with asynchronous JavaScript functions that return Promises, simplify asynchronous code with async and await, test asynchronous RESTful methods using the Jasmine framework, and perform CRUD operations on an Oracle database using Node.js.	**Promises and Testing** • Callbacks and Promises • Accessing Oracle with Node.js • Testing Asynchronous Functions
RESTful Services	9	**As a Full Stack Engineer,** develop a Single Page Application using node and Angular that utilizes RESTful services to demonstrate proficiency of skills that will be used on the job.	**The FidZulu Mini Project** • The FidZulu Mini Project
RESTful Services	10	**As a Full Stack Engineer,** understand what Apigee Edge is and learn what types of problems Apigee Edge solves to apply RESTful best practices on the job.	**Apigee Edge** • Apigee Edge
RESTful Services	11	**As a Full Stack Engineer,** understand what service virtualization is and learn what types of problems service virtualization solves to use virtual services in enterprise development.	**Service Virtualization** • Service Virtualization
RESTful Services	12	**As a Full Stack Engineer,** review and apply the basics of regular expressions to Perform Behavior-Driven Development (BDD) with Cucumber to demonstrate proficiency of skills that will be used on the job.	**Testing with Cucumber.js** • Acceptance Testing • Regular Expressions • Behavior-Driven Development with Cucumber • Testing RESTful Services

Course	Chapter	Overview	Contents
RESTful Services	13	**As a Full Stack Engineer,** apply JavaScript technologies that are integral to developing RESTful applications and demonstrate proficiency of skills that will be used on the job.	**Server-Side JavaScript Programming** • Function Techniques • JavaScript Prototype • Creating a JavaScript Factory • JavaScript Scope • Closures • Iterators and Generators • Using Grunt to Manage JavaScript Projects
RESTful Services	14	**As a Full Stack Engineer,** apply JavaScript technologies that are integral to developing RESTful applications and demonstrate proficiency of skills that will be used on the job.	**Functional and Reactive Programming in JavaScript** • Working with Arrays and Lists • Higher Order Functions • Asynchronous JavaScript • Composition • Currying • Reactive Programming

Appendix A: Building RESTful Web Services with JAX-RS

Attachment A: FSE Track – ROI Course Schedule[i]

Day	Chapters	Associate Daily Learning Topics	Time
Developing Dynamic Web Applications			
1	**1-5**	**Review (Dynamic Web)**	AM
1	6	Introduction to Angular	PM
2	7, 8	Angular Components; Angular Modules and Binding	AM/PM
3	9, 10	Angular Services; Angular End-to-End (E2E) Testing Applications	AM/PM
4	11	Build an Application	AM/PM
5	12, 13	Pipes; Angular Routing	AM/PM
6	14	Angular Forms	AM/PM
7	**15**	**Curriculum Review/Assessment (Angular)**	**AM**
7	15	Angular Deployment	PM
Programming with Java			
1	**1 - 7**	**Review (Java)**	AM/PM
2	8, 9	Abstract Classes; Java Collections	AM/PM
3	10, 11	Refactoring; Exceptions	AM/PM
4	12, 13	API Design; Design Patterns	AM/PM
4		**Assessment (Java)**	**PM**
Working with Relational Databases			
1	7, 8	Databases w/ JDBC; Testing w/Databases	AM/PM
2	9, 10	Updating Databases; Working with a Data Access Object	AM/PM
3		**Curriculum Review**	**AM**
3		**Assessment (Java with JDBC)**	**PM**
Mastering Spring and MyBatis			
1	1, 2	Introducing the Spring Framework; Understanding Spring	AM/PM
2	3, 4	Adv Spring Configuration; Introduction to MyBatis and Spring	AM/PM
3	4, 5	Introduction to MyBatis and Spring; Working Effectively with MyBatis	AM
3		**Checkpoint (Spring with MyBatis)**	**PM**
3	6	Functional Programming	PM
Developing RESTful Services with Java and Spring			
1	1, 2	Building RESTful Services; Designing RESTful Services	AM/PM
2	3, 4	Testing RESTful Services; Securing RESTful Services	AM/PM
3	5, 6	Cloud Design Patterns; Node.js	AM/PM
4	7, 8	Node.js, Express; Testing Node w/ Jasmine	AM/PM
5		**Curriculum Review and Assessment**	**AM**
5	10	Apigee Edge	PM
Developing RESTful Services (NodeJS)			
1	12, 13	BDD Using Cucumber; Server-Side JavaScript Programming	AM/PM
2	13	Server-Side JavaScript Programming; Curriculum Review	AM/PM
3	14	Functional and Reactive Programming in JavaScript	AM/PM

[i] Chapters not referenced are covered in associates self-paced learning modules (Pluralsight)