**Core Java 8 + Spring+Spring Boot**

**Lab Requirements**

**Java 8 or above**

**Intellij IDE**

**Postgres 10 or above**

**Program Duration: 8 days**

**Day 1**

**Contents**:

* **Object Orientation** 
  + Encapsulation
  + Inheritance, Is-A, Has-A
  + Polymorphism
  + Overridden Methods
  + Overloaded Methods
  + Reference Variable Casting
  + Implementing an Interface
  + Legal Return Types
  + Instance of Comparison
  + Return Type Declarations
  + Returning a Value
  + Constructors and Instantiation
  + Default Constructor
  + Overloaded Constructors
  + Statics
  + Static Variables and Methods
  + Coupling and Cohesion
* **Assignments**
  + Stack and Heap—Quick Review
  + Literals, Assignments, and Variables
  + Literal Values for All Primitive Types
  + Assignment Operators
  + Casting Primitives
  + Using a Variable or Array Element That Is Uninitialized and Unassigned
  + Local (Stack, Automatic) Primitives and Objects
  + Passing Variables into Methods
  + Passing Object Reference Variables
  + Does Java Use Pass-By-Value Semantics?
  + Passing Primitive Variables
  + Initialization Blocks
  + Using Wrapper Classes and Boxing
  + An Overview of the Wrapper Classes
  + Creating Wrapper Objects
  + Using Wrapper Conversion Utilities
  + Autoboxing
  + Overloading
  + Garbage Collection
  + Overview of Memory Management and Garbage Collection
  + Overview of Java's Garbage Collector
  + Writing Code That Explicitly Makes Objects Eligible for Garbage Collection

* **Gradle Fundamentals**
  + Introduction
  + Folder Structure
  + The pom.xml
  + Dependencies
  + Goals
  + Scopes
  + The Compiler Plugin
  + Source Plugin
  + Jar Plugin

Day 2

* **TDD with Junit 5**
  + Types of Tests
  + Why Unit Tests Are Important
  + What's JUnit?
  + JUnit 5 Architecture
  + IDEs and Build Tool Support
  + Setting up JUnit with Maven
  + Lifecycle Methods
  + Test Hierarchies
  + Assertions
  + Disabling Tests
  + Assumptions
  + Test Interfaces and Default Methods
  + Repeating Tests
  + Dynamic Tests
  + Parameterized Tests
  + Argument Sources
  + Argument Conversion
  + What Is TDD?
  + History of TDD
  + Why Practice TDD?
  + Types of Testing
  + Testing Frameworks and Tools
  + Testing Concepts
  + Insights from Testing
  + Mocking Concepts
  + Mockito Overview
  + Mockito Demo
  + Creating Mock Instances
  + Stubbing Method Calls
* **Generics and Collections**
  + Overriding hashCode() and equals()
  + Overriding equals()
  + Overriding hashCode()
  + Collections
  + So What Do You Do with a Collection?
  + List Interface
  + Set Interface
  + Map Interface
  + Queue Interface
  + Using the Collections Framework
  + ArrayList Basics
  + Autoboxing with Collections
  + Sorting Collections and Arrays
  + Navigating (Searching) TreeSets and TreeMaps
  + Other Navigation Methods
  + Backed Collections
  + Generic Types
  + Generics and Legacy Code
  + Mixing Generic and Non-generic Collections
  + Polymorphism and Generics

Day 3

* **Threads**
  + Defining, Instantiating, and Starting Threads
  + Defining a Thread
  + Instantiating a Thread
  + Starting a Thread
  + Thread States and Transitions
  + Thread States
  + Preventing Thread Execution
  + Sleeping
  + Thread Priorities and yield( )
  + Synchronizing Code
  + Synchronization and Locks
  + Thread Deadlock
  + Thread Interaction
  + Using notifyAll( ) When Many Threads May Be Waiting
* **Concurrent Patterns in Java**
  + Introducing Executors, What Is Wrong with the Runnable Pattern?
  + Defining the Executor Pattern: A New Pattern to Launch Threads
  + Defining the Executor Service Pattern, a First Simple Example
  + Comparing the Runnable and the Executor Service Patterns
  + Understanding the Waiting Queue of the Executor Service
  + Wrapping-up the Executor Service Pattern
  + From Runnable to Callable: What Is Wrong with Runnables?
  + Defining a New Model for Tasks That Return Objects
  + Introducing the Callable Interface to Model Tasks
  + Introducing the Future Object to Transmit Objects Between Threads
  + Wrapping-up Callables and Futures, Handling Exceptions
* **Concurrent Collections**
  + Implementing Concurrency at the API Level
  + Hierarchy of Collection and Map, Concurrent Interfaces
  + What Does It Mean for an Interface to Be Concurrent?
  + Why You Should Avoid Vectors and Stacks
  + Understanding Copy On Write Arrays
  + Introducing Queue and Deque, and Their Implementations
  + Understanding How Queue Works in a Concurrent Environment
  + Adding Elements to a Queue That Is Full: How Can It Fail?
  + Understanding Error Handling in Queue and Deque
  + Introducing Concurrent Maps and Their Implementations
  + Atomic Operations Defined by the ConcurrentMap Interface
  + Understanding Concurrency for a HashMap
  + Understanding the Structure of the ConcurrentHashMap from Java 7
  + Introducing the Java 8 ConcurrentHashMap and Its Parallel Methods
  + Parallel Search on a Java 8 ConcurrentHashMap
  + Parallel Map / Reduce on a Java 8 ConcurrentHashMap
  + Parallel ForEach on a Java 8 ConcurrentHashMap
  + Creating a Concurrent Set on a Java 8 ConcurrentHashMap
  + Introducing Skip Lists to Implement ConcurrentMap
  + Understanding How Linked Lists Can Be Improved by Skip Lists
  + How to Make a Skip List Concurrent Without Synchronization

**Database & SQL**

**Day 4**

**Contents:**

* **Introduction**
  + The Relational Model
* **Understanding Basic SQL Syntax**
  + The Relational Model
  + Basic SQL Commands - SELECT
  + Basic SQL Commands - INSERT
  + Basic SQL Commands - UPDATE
  + Basic SQL Commands – DELETE
* **Querying Data with the SELECT Statement**
  + The SELECT List
  + SELECT List Wildcard (\*)
  + The FROM Clause
  + How to Constrain the Result Set
  + DISTINCT and NOT DISTINCT
* **Filtering Results with the Where Clause**
  + WHERE Clause
  + Boolean Operators
  + The AND Keyword
  + The OR Keyword
  + Other Boolean Operators BETWEEN, LIKE, IN, IS, IS NOT
* **Shaping Results with ORDER BY and GROUP BY**
  + ORDER BY
  + Set Functions
  + Set Function And Qualifiers
  + GROUP BY
  + HAVING clause
* **Creating Database Tables**
  + CREATE DATABASE
  + CREATE TABLE
  + NULL Values
  + PRIMARY KEY
  + CONSTRAINT
  + ALTER TABLE
  + DROP TABLE

Day 5

* **Lambda Expressions**
  + Introduction
  + Writing Lambda Expressions
  + Functional Interfaces
  + Types of Functional Interfaces
  + Method reference
* **Stream API**
  + Introduction
  + Stream API with Collections
  + Stream Operations

**JDBC**

* **Introduction to JDBC**
  + Introduction to JDBC
  + Architecture of JDBC
  + Role of Driver Manager
  + Understanding JDBC Driver Types
* **Getting Started with JDBC**
  + Connecting to Database using JDBC
  + Managing Database resources
  + Handling JDBC Exceptions
* **Performing Basic CRUD operations using JDBC**
  + Executing Static SQL Statements
  + Iterating Through ResultSets
  + Understanding Scrollable ResultSets
  + Understanding Updatable ResultSets
  + Understanding PreparedStatement
  + Retrieving Data Using PreparedStatement
  + Inserting the Record
  + Updating the Record
  + Removing the Record

Day 6

**Spring +Spring Boot**

**Program Duration- 7 Days**

**Contents:**

* Introduction to ORM and its need
* The Persistence Life Cycle
* Java persistence API (JPA)
* JPQL

Association and Mapping

|  |
| --- |
| * Spring core Introduction Singleton factor design pattern |
| * Spring - Dependency Injection example with XML cfiguration |
| * Spring - Instantiating beans example |
| * Spring - Hello world example with annotationon |
| * Spring - Hello world example |
| * Spring - @PropertySource annotationon example |
| * Spring - Custom qualifier annotation example |
| * Spring - @Qualifier annotation example |
| * Spring - Auto Compent scanning example |
| * Spring - @Import and @ImportResource example |
| * Spring - @Primary annotation example |
| * Spring - @Resource annotation example |
| * Spring - @Required annotation example |
| * Spring Bean Life Cycle - @Bean’s initMethod and destroyMethod attributes example |
| * Spring Bean Life Cycle - @PostCstruct and @PreDestroy example |
| * Spring - Prototype scope example using @Scope annotation |
| * Spring - Singlet scope example using @Scope annotation |
| * Spring - @Lazy annotation example |
| * Spring - @Depends annotation example |
| * Spring - Injecting Collections into bean example |
| * Spring - Injecting Properties into bean example |
| * Spring - Cstructor Injection and arguments ambiguities |
| * Spring - Dependency Injection example with annotation |
| * Spring @Profiles Example |
| * Spring JDBC template |
| * Spring Data JPA |
| * When to use Spring Data JPA? |
| * Spring Data JPA Example |
| * Spring JAP Example Sample Data |
| * Spring Data JPA Maven Project Structure |
| * Spring Data JPA Maven Dependencies |
| * Spring Configuration Classes |
| * Model Class |
| * Spring Data JPA Repository |
| * Spring Service Class |
| * Spring Controller Class |

Day 7

**SpringBoot**

|  |
| --- |
| Spring Boot – Starter parent |
| Spring Boot – Starter templates |
| Spring Boot – Multi-module project |
| Spring Boot – Annotations |
| Spring Boot – Auto configuration |
| Spring Boot – AOP |
| Spring Boot – Logging |
| Spring Boot – DevTools |
| Spring Boot – WAR Packaging |
| Spring Boot – REST API |
| Spring Boot – Caching |
| Spring Boot – Retry |
| Spring Boot – CommandLineRunner |

**Day 8**

|  |
| --- |
| What is MongoDB? |
| Join our Online Learning Community |
| The Key MongoDB Characteristics (and how they differ from SQL Databases) |
| Understanding the MongoDB Ecosystem |
| Shell vs Drivers |
| MongoDB + Clients The Big Picture |
| Understanding the Basics & CRUD Operations |
| Module Introduction |
| Understanding Databases, Collections & Documents |
| The Shell & MongoDB Drivers for Different Languages |
| Creating Databases & Collections |
| Understanding JSON Data |
| Comparing JSON & BSON |
| 1. MongoDB Java CRUD Example for Beginners |
| 2. MongoDB insert |
| 3. MongoDB Bulk Insert |
| 4. MongoDB update |
| 5. MongoDB find |
| 6. MongoDB findOne |
| 7. MongoDB findAndModify |
| 8. MongoDB distinct |
| 9. MongoDB exists Example |
| 10. MongoDB remove Example |
| 11. MongoDB upsert Example |
| 12. MongoDB sort |
| 13. AND OR LIKE OPERATOR BASED QUERY |
| 14.GET DOCUMENT BY SUBDOCUMENT |
| 15. GET DOCUMENT BY ARRAY |
| 16. PAGINATION |
| 17. BULK INSERT |
| 18. MAP REDUCER |