LTI
Let's Solve

Celebrating 20 Years

Java Full Stack Course Content

A Larsen & Toubro Group Company

Objectives of this Course:

- To understand the role of Database in our applications
- To understand the role of Java technology in modern application development
- To understand how to develop web applications that are scalable, maintainable
- To understand the architecture and design of web applications
- To understand modern development techniques using JS Frameworks like Angular
- To understand how to separate the application concerns based on functionality
- To understand role of Spring framework for managing business processes
- To understand effective and clean division between controllers, models and view using Spring MVC
- To understand Data Persistence using Hibernate Framework
- To understand importance of DevOps & various tools like Git, Maven, Jenkins
- To understand Agile methodology & Scrum software development practices

Prerequisites:

- Basic Knowledge of Programming Techniques, Database & SDLC

Course Outline

Boot Camp Phase

Communication Training (3 Days)

- Spread across 3 weeks / 1 day per week

Day 1 – 3:

DBMS – DBMS Concepts and SQL

- Introduction to Databases
- Database Models
  - Relational Model
- Data Design and Normalization
- Structured Query Language and its categories
  - DDL – DML – DQL – DCL – TCL
- SELECT statement varieties with clauses
  - WHERE clause
  - GROUP BY clause
  - HAVING clause
  - ORDER BY clause
- Using Oracle built in Functions
- Joining the tables – Join variants
  - Equi and Non-Equi Joins
  - Self-Join
  - Cartesian Product
  - Outer Join
- Subqueries
- Implementing Views
- Implementing Data Integrity by using Constraints
  - Data Integrity Overview
  - Creating Constraints
  - Implementing Constraints
  - Not Null
  - Unique Key
  - Primary key
  - Check Constraints
  - Default
  - Foreign Key
- Implementing Stored Procedures and Functions

Day 4 - 9:

Java Basics

- Describe the features of Java
- Describe the real-world applications of Java
- Describe the Java Development Kit (JDK) and the Java Runtime Environment (JRE)

- Describe the components of a basic Java program
- Compile and execute a Java program

Working with Java Data Types

- Declare and initialize variables (including casting of primitive data types)
- Differentiate between object reference variables and primitive variables
- Develop code that uses wrapper classes such as Boolean, Double, and Integer
- Declare, instantiate, initialize and use an array

Java Flow Control

- Branching
- Switching
- Looping
- Using Foreach

Object Oriented Programming

- Characteristics of Object-Oriented Programming
- Classes and Objects
- Packages
- Constructor
- Properties
- Inheritance
- Access Modifiers
- Abstract classes and methods
- Static, this, super and final keyword
- Interfaces

Exception Handling

- Checked vs. Unchecked Exceptions
- Built in Exceptions
- Handling Exceptions
- Custom Exception classes
- Throwing exceptions

Working with Selected classes from the Java API

- Manipulate data using the StringBuilder class and its methods
- Create and manipulate Strings
- Using Java SE 8 Date/Time API

Generics and Collections

- Create and use a generic class
- Create and use List, Set and Map
- Sorting data in collection
- Collections Streams and Filters
- Iterate using forEach methods of Streams and List

Building Database Applications with JDBC

- Describe the interfaces that make up the core of the JDBC API including the Driver, Connection, Statement, and ResultSet interfaces and their relationship to provider implementations
- Identify the components required to connect to a database using the DriverManager class including the JDBC URL
- Submit queries and read results from the database including creating statements, returning result sets, iterating through the results, and properly closing result sets, statements, and connections

Mini Project + Mock Client Interview

Day 10:
Java Codility

Introduction to Data Structures & Algorithms

- Introduction to Codility Platform
- Understanding Computational Thinking
- Understanding Space and Time Complexity
- Understanding Big-O notation
- Algorithm Run Time Analysis

Coding Problems and Challenges:

- Iterations & Arrays
  - Iteration techniques for arrays and collections
  - Solving coding challenges on Arrays
- Time Complexity
  - Revisiting Big-O notations
  - Writing efficient algorithms to improve performance
- Counting Elements
  - Understanding counting of elements algorithm
  - Using this technique for solving various problems

Core Java Codility Test

Day 11:
Web Technologies – HTML

- Understanding & using HTML5

Web Technologies - CSS 3

- CSS3 Introduction and commonly used CSS 3 properties

Web Technologies – JavaScript

- Introduction to JavaScript
- JavaScript Events and Functions
- JavaScript Form Validation

Day 12-15:
Advanced Web Technologies – Angular 10

- Angular - Introduction
- Understanding Single Page Applications (SPA)
- AngularJS 1.x vs Angular recent versions
- Introduction to TypeScript
    - Role of typescript in Angular
- Developing a simple Angular application
- Writing custom components
- Understanding One-way data binding
- Understanding Two-way data binding
- Angular forms and it's types
- Form validation
- Angular Routing and DI (Dependency Injection)

Mini Project + Assessment + Angular Codility Test

Specialization Phase

Day 16 - 18:
DevOps – Overview of Dev-Ops

- What is DevOps
- Continuous Integration
- Continuous Deployment

GIT: Version Control

- Introduction to Git and Github
- About Version Control System and Types
- GIT Basics
- GIT Command Line
- Creating repository
- Cloning, check-in and committing
- Fetch pull and remote
- Branching
    - Creating, switching and merging branches

Maven

- Maven Installation
- Maven Basics
- Maven Demo
- Maven Eclipse Demo
- Maven lifecycle
- Maven Repositories

JUnit 5

- Overview
- Unit Testing and JUnit Overview

- New Features in JUnit 5
- JUnit 5 Library Components
- Naming Conventions and Organizing Tests
- Writing Test Methods
- Assertions

Jenkins – Continuous Integration / Continuous Delivery/Deployment (CI/CD)

- Understanding CI/CD
- Introduction about Jenkins
- Build Cycle
- Jenkins Architecture
- Installation
  - Installing and configuring Jenkins
- Exploring Jenkins Dashboard
- Jobs
  - Creating Jobs
  - Running the Jobs
  - Setting up the global environments for Jobs
- Adding and updating Plugins
- Disabling and deleting jobs

Day 19 – 21:

Introduction to Hibernate Framework

- Object Persistence
  - Object/Relational Paradigms
  - O/R Mismatch
  - Object Relational Mapping (ORM)
  - Java ORM/Persistent Frameworks
- Hibernate Architecture
  - Hibernate Architecture and API
- Introduction to JPA
  - Hibernate and JPA relationship
  - Why JPA?
- Using Hibernate/JPA
  - Using annotations for persistence mapping
  - Configuration of the persistence context
  - Understanding EntityManagerFactory
  - Working with EntityManager for persisting objects
  - Transactions and flushing
- Persistent Classes
  - POJOs
  - JavaBeans - Basic Mapping
  - Class to Table Mappings - Property Mapping
  - Identifiers – Generators
- Working with Persistent Objects
  - Entity Lifecycle
  - Transient State - Persistent State

        - Persistent Object State
        - Detached State
        - Object Identifiers Revisited
    - HQL/JPQL
        - Introduction
        - Difference between JPQL and SQL
        - Handling joins and common other cases
        - Named queries

Day 22-25:
Introduction to Spring Framework

- Spring Overview
    - Spring Philosophy
    - Spring Architecture and Modules
- A First Spring Application
    - Dependency Injection (DI)
        - Spring Containers
        - BeanFactory
        - ApplicationContext
        - Bean Definitions & Property Injection
    - Understanding @Component and it's types
    - Using @Autowired for DI
    - Bean Lifecycle

Database support in Spring

- Spring support for DataSource
    - Configuration
    - Benefits of using DataSource
        - Connection Pooling advantage
- Introduction to JdbcTemplate API
    - Performing insert/update/delete
    - Handling select queries
        - Role of RowMapper
- Hibernate/JPA integration with Spring
    - Benefits of using Spring on top of Hibernate/JPA
    - Steps for integration
    - Transaction support in Spring

Spring MVC

- Spring MVC Architecture
- Spring MVC Configuration and Setup
- DispatcherServlet
- Context Configuration
- Annotation Based Controllers
- Understanding the flow
- Commonly used annotations in Spring MVC
- Spring MVC for REST

- Understanding support for REST WebServices
- Writing @RestController
- Handling incoming/outgoing JSON
- Implementing GET and POST calls

Spring Boot Introduction

- Introduction to Spring Boot
- Value Proposition of Spring Boot
- High-level Spring Boot features
- Creating a simple Boot application using Spring Initializr web-site

Spring Boot Dependencies, Auto-Configuration & Runtime

- Dependency management using Spring Boot starters
- How auto-configuration works
- Configuration properties
- Overriding auto-configuration

JPA with Spring and Spring Data

- Quick introduction to ORM with JPA
- Benefits of using Spring with JPA
- JPA configuration in Spring
- Configuring Spring JPA using Spring Boot
- Spring Data JPA dynamic repositories

Assessment

Project Gladiator Phase

Day 26-31:
Project Gladiator
Project Gladiator Evaluation