Java Fullstack Program Curriculum

DLithe

Be Agile..Be in Demand

Training sessions will be comprised of both activities and lectures that will touch upon the following levels

- L0 - Core Java, React and JDBC
- L1 - SpringBoot, CRUD operations, Web Application Development

Program Overview: Fullstack

This FullStack Java Certification Training Course will help you master front-end web development with React and back-end development with Java and SpringBoot. Gain in-depth knowledge of React concepts such as facilitating the development of single-page web applications, dependency injection, typescript, components, directives, pipes, forms, routing, and testing. Also, Java, SpringBoot concepts such as java fundamentals, object oriented programming, classes, methods, multithreading concepts, exception handling, collection framework, and JDBC.

Course Flow

The expert shall conduct training in FullStack Java development in accordance with the course content mentioned below..

DLithe
Be Agile..Be in Demand

Course Content

L0 - Core Java, Web Fundamentals, React, and JDBC

- Java Language Environment
- Java Fundamentals
- Essentials of Object-Oriented Programming
- Writing Java Classes
- Packages
- Exception Handling
- I/O Operations in Java
- Collections Framework
- Multithreaded Programming
- Web Fundamentals
- Introduction to React and its concepts
- Java JDBC

L1 - SpringBoot, CRUD operations, Web Application

- Introduction to spring boot
- Building Spring Boot Application
- Rest Annotation with In Memory Database & CRUD Operations
- JPA Repository Concepts
- Actuator Concepts
- Auto Configuration
- Microservices

Detailed Curriculum:

L0

- **Java Language Environment**

  In this Module you will learn what java is, and its features, and why it is popular?

  - Object Oriented
  - Platform Independent
  - Automatic Memory Management
  - Compiled / Interpreted approach
  - Robust
  - Secure
  - Dynamic Linking
  - Multi-Threaded

- **Java Fundamentals**

  In this module you will learn the basic structure of the programming and how to create your own structural code, and where to use it.

  - Data types
  - Operators
  - Control Statements
  - Arrays
  - Enhanced for-loop
  - Enumerated types
  - Static import
  - Auto boxing
  - Formatted I/O
  - Variable arguments

- **Essentials of Object-Oriented Programming**

  In this module you will learn the basic definitions and uses and how to make our code in a more structured way,so that anyone can understand our code, how to make it easier.

  - Object and Class Definition
  - Using Encapsulation to combine methods and data in a single class
  - Abstraction
  - Inheritance
  - Polymorphism

- **Writing Java Classes**
  In this module you will learn all the concepts Oops where we will use all these concepts in our daily life by knowingly or unknowingly. By learning this module you are able to create a code in a standard format.

  - Encapsulation
  - Polymorphism
  - Inheritance
  - OOP in Java
  - Class Fundamentals
  - Using Objects
  - Constructor
  - Garbage Collection
  - Method Overloading
  - Method Overriding
  - Static Members
  - Understanding Interface
  - Using Interfaces

- **Packages**

  In this module you will learn how to re-use/access our class files when it is in the same package/different package/different project.
  - Why packages
  - Understanding Class path
  - Access modifiers & their Scope

- **Exception Handling**

  In this module you will learn how to handle our standalone applications/web applications, whenever an error occurs, how to tackle it, and where it is occurring, by learning this module you will get it.

  - When an exception occurs.
  - Importance of Exception Handling
  - Exception Propagation
  - Exception Types
  - Using try and catch
  - Throw
  - Throws
  - Finally
  - Writing User defined Exceptions

- **I/O Operations in Java**

  In this module you will learn how to create a file and how to modify/read/write/handle an existing file, through your code and you can make your file access permission rights.

  - Byte Oriented Streams
  - File Handling
  - Readers and Writers

- **Collections Framework**

  In this module you will learn how to make/get our content in a user's perspective/his requirement, when it is in the same file or it may be in a different file even if it is in different format.

  - Collection & Iterator Interface
  - Enumeration
  - List, ArrayList and Vector
  - Comparator and Comparable
  - Set Interface & SortedSet
  - Hashtable
  - Properties

- **Multithreaded Programming**

  In this module you will learn how to perform multiple tasks at a same time or it may be partially. Here tasks can be either running multiple code simultaneously when some background code is running or to run the code one after another or it may be at a time.

  - Introduction to Multithreading
  - Understanding Threads & its States
  - Java Threading Model
  - Thread class & Runnable Interface
  - Thread Priorities
  - Thread Synchronisation
  - Inter Thread Communication

- **Web Fundamentals**
  - Getting started with the web
  - Installing basic software
  - What will your website look like?
  - Dealing with files
  - HTML basics
  - CSS basics
  - JavaScript basics
  - Publishing your website
  - How the web works
  - Introduction to HTML
  - Multimedia and embedding
  - HTML tables
  - CSS first steps
  - CSS building blocks
  - Styling text
  - CSS layout
  - JavaScript first steps
  - JavaScript building blocks

- Introducing JavaScript objects
- Asynchronous JavaScript

- **React**
  - Learn what are the stateful and stateless components and when to use them
  - Working with function based and class based components
  - Working with React Modules, importing and exporting the modules
  - Learn in detail about how the render method works
  - React component lifecycle and different lifecycle methods
  - Creating a proper working structure for a project from scratch
  - Creating dynamic websites with help of re-usable components

- **Java JDBC**

  In this module you will learn how to access tabular data stored in any relational database and perform database operations like fetch, update, delete and add.

  - Types of JDBC Drivers
  - Create a JDBC connection class
  - What is JDBC API

L1

- **Introduction to SpringBoot**

  - Types of Software Architecture
  - SOA and Monolith Architecture
  - Need of SpringBoot
  - Difference between Spring and SpringBoot
  - Advantages of Microservices

- **Building SpringBoot Application**

  - Using spring initializer
  - Converting maven project to spring boot
  - Using IntellIJIdea

- **Rest annotations with in-memory DB and CRUD operations**

  - H2 Database
  - Accessing H2 database
  - DB Configuration
  - Postman

- **JPA Repository Concepts**
  - CRUD Repository
  - JPA Query concepts
  - Named Queries
  - Query Annotations
  - Pagination and Sorting

- **Actuator Concepts**
  - Implementation of Actuator
  - Health check
  - Production monitoring

- **Auto Configurations**

  - Introduction
  - SpringBoot built in conditional annotations
  - Customise conditional annotations

- **Microservices**

    - Introduction
    - Advantages
    - Disadvantages
    - Examples

- **Dev-Ops**

    - CI/CD : Git
    - Agile Development
    - Overview on Domain and Process
    - Job Assistence

Expected Outcome:Project

Build an Ecommerce web application by implementing CRUD functionality in Angular and build an Angular app with the following features: Product form, product list, and view, delete, and update products from the list. Also, provide suitable backend APIs to integrate with Front-end features.