FUSION
SOFTWARE INSTITUTE

JAVA FULL STACK DEVELOPMENT

Java Full Stack Development Course:
Your Pathway to Excellence

www.fusion-institute.com

INTRODUCTION

In today's dynamic tech landscape, mastering Java Full-Stack Development is the key to building powerful, scalable, and secure applications that drive innovation across industries. At Fusion Software Institute, our Java Full-Stack Development Course provides a comprehensive curriculum for beginners and experienced developers.

What You'll Learn?

- Core Java Concepts: Master object-oriented programming, data structures, and algorithms.
- Frontend & Backend Integration: Build dynamic user interfaces with frameworks like Angular or React.
- Frameworks Expertise: Dive deep into Spring Boot and Hibernate for seamless application development.
- Database Management: Learn how to manage and optimize databases efficiently.
- Real-World Projects: Work on industry-relevant projects to apply your skills practically.

Career Opportunities

This course equips you for high-demand roles like:

- Java Developer
- Full Stack Developer
- Software Engineer
- Backend Developer

Why Choose Java Full Stack?

Java powers the backbone of global industries from e-commerce to banking, healthcare, and beyond. As demand for skilled Java developers grows, this course prepares you to thrive in a competitive market.

Begin your journey with Fusion Software Institute and transform your passion for coding into a successful career. Learn, Code, and Lead with Java Full Stack!

INTRODUCTION TO JAVA

- Introduction to Java
- History of Java
- Important Features of Java
- JVM Architecture
- JDK 1.8, JRE and JVM
- Java Keywords
- Simple Hello World Program
- Java Flow Control
- Class and Objects
- Data Types

- Primitive Data Types
- Non – Primitive Data Types
- Constructor
- Instance & Static Variable
- Static Block
- Has-A Relation (secondary Reference)
- Setter and Getter
- Method return type
  (primitive type and class type)

LOGIC BUILDING

- Operator
- Java ternary operator if, if else
- Switch Statement
- Conditional Related Problem Looping Control
- Java for loop
- Java while loop
- Java do while loop

- Java break Statement
- Java continue statement
- Looping Related Problems
- Array

OBJECT ORIENTED PROGRAMMING (OOPS)

- **Encapsulation**
- Access Modifier Data Hiding
- Protected Access Modifier Implementation
- **Inheritance (IS-A)**
- Constructor and Inheritance
- Inheritance with parent reference and Child constructor
- Covariant Return Types
- Inheritance and type casting
- This/Super Keyword

- **Polymorphism**
- Method Overloading
- Constructor Overloading
- Compile time Polymorphism
- Overloading and Narrowing Concept Method Overriding
- @override Annotation
- Overriding

- **Abstraction**
- Abstract class
- Interface
- Marker Interface
- Interface & Multiple Inheritance
- Interface Uses and Benefit with example
- Difference of Interface and Abstract class

STRING HANDLING

- String: What and Why?
- Immutable String
- String Comparison
- String Concatenation
- Substring, Methods of String class
- StringBuffer class, StringBuilder class
- Creating Immutable class
- String Tokenizer

IMPORTANT CONCEPTS & CLASSES:

- Package
  - Package and import keyword
  - Access modifiers: public, private, protected
- Object Class
  - hashCode(), toString(), equals()
  - clone(), finalizable(), getClass()
- Wrapper Classes

EXCEPTION HANDLING

- What and Why?
- Try and catch block
- Multiple catch blocks
- Nested try
- Finally block
- Finally block with return statement
- Throw keyword
- Exception Propagation
- Throws keyword
- Throws keyword with Method Overriding
- Throws keyword with Constructor
- Custom Exception
- jdk 1.7 Feature AutoCloseable Interface
- jdk1.7 Feature try with Resources

MULTITHREADING

- Introduction
- Extends Thread Class
- Implementing Runnable interface and Callable interface
- Thread Life Cycle
- Demon Thread & Non-Demon Thread
- Locking System
- Inter Thread Communication
- Dead Lock System
- Thread Methods (sleep, join, yield etc.)
- Synchronization: What and Why?
- Synchronized method synchronized block
- Static synchronization Deadlock
- Inter-thread Communication

COLLECTION FRAMEWORK

- Java Collection
- Java List : ArrayList, Vector, LinkedList, Sorting
- Java Set : HashSet, TreeSet, LinkedHashSet
- Comparable & Comparator
- Java Map : HashMap, TreeMap, LinkedHashMap
- Iterator, List Iterator, Enumeration
- Java Collections (Utility Class)

INPUT AND OUTPUT

- FileOutputStream & FileInputStream
- BufferedOutputStream & BufferedInputStream
- DataInputStream & DataOutputStream
- Input from keyboard by InputStreamReader
- Input from keyboard by Console
- Input from keyboard by Scanner Class
- ObjectInputStream & ObectOutputStream
- Serialization & Deserialization
- Serialization with IS-A and Has-A
- Transient keyword
- FileWriter & FileReader

SQL

- Database Introduction
- SQL Introduction, MYSQL Introduction
- MYSQL Installation, MYSQL Workbench Installation
- Features, Data Types, DDL, DML, TCL
- Create Table, Select Statement, Insert Query
- Delete Query, Update Query
- Constraints, DISTINCT Clause, WHERE Clause
- MYSQL Conditions : AND, OR, BOOLEAN, LIKE, IN
- MYSQL Functions : MIN, MAX, AVG, SUM, COUNT
- ORDER BY Clause, GROUP BY Clause
- SQL JOINS :
  - SQL Outer Join,
  - SQL Left Join
  - SQL Full Join
  - SQL Inner Join
  - SQL Right Join
- Stored Procedure

ADVANCED JAVA

- JDBC
- JDBC Drivers
- Steps to connect to the database
- Connectivity with MySQL
- Driver Manager, Connection interface
- Statement interface, ResultSet interface
- PreparedStatement, Callable Statement
- ResultSetMetadata, DatabaseMetadata
- Transaction Management
- CRUD Operations

WEB APPLICATIONS

- What is web application?
- What is web server?
- Difference between web server & application server

- **Servlet :**
- Different ways to design servlet
- Servlet Life Cycle Stages, Life cycle methods
- ServletRequest, ServletResponse
- GET and POST request
- sendRedirect, RequestDispatcher : include(), forward()
- web.xml : Deployment Descriptor, @WebServlet
- ServletConfig, ServletContext
- Session Management : Cookie, HttpSession

- **JSP:**
- JSP Life cycle stages, Life cycle methods
- Implicit objects in JSP
- Basic building blocks :
  - Scripting elements
  - Directive elements
  - Action elements
- JSTL Core Tags, EL

FULLSTACK & ADVANCE FRAMEWORK

- **HIBERNATE (Hibernate 5.X)**
- Introduction
- ORM (Object Relational Mapping)
- Advantage of ORM
- Hibernate architecture
- JPA, Hibernate with JPA
- Hibernate Advantages over JDBC
- Hibernate Configuration File & Annotation
- Hibernate with Java Based (Zero XML file)
- Load and get method difference
- save(), saveorUpdate() , persist() method
- Hibernate Mapping(Has-A)
- one-To-One, One-To-Many, Many-To-One, Many-To- Many
- Hibernate Query Language(HQL)
- Hibernate Annotations

- Caching in Hibernate : First Level Cache
- Hibernate Transaction Management
- CRUD Operation

SPRING

- **Spring IOC**
- What is Spring?
- Spring Modules
- What is IOC?
- Spring IOC Container
- Bean Factory/Core Container
- Application Context/Advance or J2EE Container
- Spring Bean life Cycle
- Bean Scope : Singleton Scope, Prototype Scope
- Lazy and Eagar Loading Concept
- Dependency Injection: Setter Based, Constructor Based

- **Spring Web MVC**
- Spring Web MVC
- Features Of Web MVC
- Life Cycle of Web MVC
- Execution Flow of MVC
- Configure DispatcherServlet & viewResolver
- Stereotype Annotation in MVC : @Component, @Controller, @Service,
- Spring MVC – Multiple Controllers
- RequestMapping – Working with Parameters @RequestParam, @PathVariable
- Handler Mapping, Controller Class
- View Resolvers, Form Handling
- Spring Form Validation
- **Mini Project**

SPRING BOOT

- **Spring Boot**
- Spring Boot Introduction
- Spring Boot Features
- Advantages Over Spring Web MVC
- Creating Spring Boot Application Using Maven
- Using Spring Initializer (http://start.spring.io/)
- Using Spring STS IDE
- Removal Of XML Files
- Simple Application Using Spring Boot
- Spring Boot Starters
- @SpringBootApplication Annotation
- Spring Application Class
- Embedded Servlet container

- **Spring Boot RESTful Web Services**
- SpringBoot REST Introduction
- Creating and Running Spring Boot Applications
- Building RESTful Web Service
- HTTP Methods and Status Codes

- **Spring Boot JPA**
- What is Spring Boot JPA?
- JPA Annotations
- Curd Repository, JPA Repository
- Custom Queries :
  - Using Method
  - JPA Named Queries
  - Query Annotation
- Spring Boot with Data JPA (CRUD Repository)
- Mini Project Using Spring Boot

MICROSERVICES & SPRING CLOUD

- **Microservices & Spring Cloud**
- Introduction to Microservices
- Microservices & Monolithic difference
- Microservices Architecture
- Service Discovery
  (Eureka Client & Eureka Server)
- **Mini Project: Microservices using**

REACT JS

- HTML
  - Introduction to HTML
    - What is HTML (Hypertext Markup Language)?
    - Setting up a text editor and web browser for
    - Creating your first HTML document
    - Document structure and elements
    - Text editors and Integrated Development Environments (IDEs

HTML DOCUMENT STRUCTURE

- HTML elements, tags, and attributes
- Headings, paragraphs, and line breaks
- Lists: ordered lists
- unordered lists and list items
- Hyperlinks: creating links to other web pages (element)
- Linking to external resources (images, stylesheets, & scripts)
- Semantic HTML

TEXT FORMATTING & MULTIMEDIA TEXT FORMATTING

- , , <u>, <s>, and <mark>
- Creating and formatting tables: <table>, <tr>, <td>, <th>, <caption>
- Adding images: <img> element and attributes

FORMS AND INPUT ELEMENTS CREATING FORMS:

- <form> element and its attributes
- Form controls: text input, password input, textarea,checkboxes, radio buttons, & select dropdowns
- Validating user input with HTML5 attributes (required, pattern, etc.)

CSS

HTML & CSS INTEGRATION

- Introduction to CSS
- Inline VS Interval VS External CSS
- Applying CSS styles to HTML elements using Selectors
- CSS Properties - color, margin, padding

STYLING TEXTS & FONTS

- Font properties - font-style, font-weight, font-family etc
- Text properties: color,text-align, text-decoration, & text-transform
- Google Fonts and their integration

COLORS & BACKGROUNDS

- CSS color values: hexadecimal, RGB, RGBA, HSL, and named colors
- Background properties: background-color

BOX MODEL & LAYOUT

- Understanding the CSS box model
- Margin, border, padding, and content areas
- Box-sizing property Display property and its values (inline, block, inline-block, flex, grid)
- Controlling element dimensions (width and height)

POSITIONING & LAYOUT TECHNIQUES

- Positioning schemes: static, relative, absolute, fixed, & sticky
- CSS floats and clearing floats
- CSS positioning and stacking contexts
- Centering elements horizontally and vertically
- CSS Flexbox for responsive layouts along with flex properties - justify-content and align-items and flex-direction. In-depth exploration of CSS Flexbox for advanced layouts
- Advanced CSS Grid layout techniques

RESPONSIVE WEB DESIGN WITH CSS

- Introduction to responsive design principles
- Media queries and breakpoints
- Creating responsive layouts with CSS Grid
- Fluid typography & responsive images

JAVASCRIPT

BASICS OF JAVASCRIPT

VARIABLES & DATA TYPES

- Variables: Used to store data. “var” vs “let” vs “const”
- Data Types: Include numbers, strings, booleans, arrays, objects, & more.

OPERATORS

- Arithmetic: +, -, *, /, %.
- Comparison: ==, ===, !=, !==, <, >, <=, >=.
- Logical: && (AND), || (OR), ! (NOT).
- Ternary Operator

CONTROL FLOW

- Conditional Statements: if, else if, else.
- Switch Statements: Used for multi-case branching.
- Loops - for loop: Repeats code for a specified number of Iterations. while loop: Repeats code while a condition is true. do...while loop: Similar to while loop, but code is executed at least once.

FUNCTIONS

- Blocks of code that can be called with arguments & can return a value.
- Types of functions: Arrow, Constructor, IIFE, anonymous functions.

ARRAYS

- Ordered collections of data.
- Arrays methods : pop(),push(),shift(),unshift() etc
- Higher Order functions: Map, filter, reduce.

OBJECTS

- Key-value pairs used to represent structured data.
- Accessing data : dot notation, bracket notation,
- Creating new key-values

SCOPE AND CLOSURES

- Variables have different levels of visibility (scope).
- Lexical Scope
- Hoisting
- Functional scope, block scope & global scope.
- Closures allow inner functions to access variables from outer functions.

DOM MANIPULATION

- Interacting with the Document Object Model (DOM)
- to manipulate web pages.
- DOM methods
- DOM Events

EVENTS

- Handling user interactions or actions on a web page
- Event Propagation: bubbling & capturing

ASYNCHRONOUS JAVASCRIPT

- Handling async javascript
- Event loop
- Async await.
- Try catch
- Promises: promises.all(), promises.race(), promises.any()
- Fetching data from APIs

ES6 Features

- Block scope - var, let and const
- Template Literals
- Arrow functions
- Spread and Rest operators
- Destructuring
- Classes in JavaScript

Live Project

REACT

INTRODUCTION TO REACT

- What is React ?
- What is virtual DOM ?
- Development setup Create-react-app
  Creating your first react app.
- What is JSX ?
- React children
- React Fragments
- Conditional Rendering
- List & Keys
- State vs props
- What is Props drilling ?
- Controlled & Uncontrolled Components

HOOKS IN REACT

- useState
- useEffect
- useContext
- useReducer

Handling API Data in React

Live Projects