

# Java Full Stack Course Syllabus

We Offer Comprehensive Java Full Stack Training with Industry-Standard Practices



**FULL STACK  
DEVELOPMENT**

## CORE JAVA (J2SE)

---

### Chapter 1: Introduction

- Introduction to Operating System
- Introduction to Programming Language
- What is Java
- History of Java
- Types of Applications
- Release Versions and Installation of Java/JDK
- Creating Java Files
- Compilation and Execution of Java Program

---

### Chapter 2: Classes and Objects

- What is a Class
- Internal Details of a Class
- What is an Object
- Object Creation / Instantiation
- Reference/Object/Instance Name
- Execution of Programs with Objects
- Static (Class) Variables
- Non-static (Instance/Object) Variables
- Local Variables
- Memory Allocation for Variables

---

### Chapter 3: Methods and Data Types

- Static Keyword & Functionality
  - About `main(String[] args)`
  - Predefined and User-Defined Methods
  - Memory Allocation for Methods
  - Data Types: Primitive & User-Defined
  - Static / Non-Static Methods
  - Return Types and Values
  - Method Parameters & Arguments
-

## Chapter 4: Eclipse & Packages

- Creating Workspace and Project in Eclipse
  - String Class and Methods
  - Comments in Java
  - Multiple Classes in a Single Source File
  - Keywords and Packages
  - Creating and Importing Packages
  - Interactive Programs using Scanner Class
  - Naming Conventions
- 

## Chapter 5: Operators

- Operators & Operands
  - Assignment, Arithmetic, Unary, Conditional
  - Relational, Logical, Bitwise, and Ternary Operators
- 

## Chapter 6: Control Statements

- if, if-else, if-else-if, Nested if
  - switch
  - for, while, do-while
  - break, continue
- 

## Chapter 7: Constructors

- What is a Constructor
  - Default & Parameterized Constructors
  - Difference between Constructor and Method
- 

## Chapter 8: Arrays

- What is an Array
  - Single & Multi-Dimensional Arrays
  - Enhanced for-each Loop
  - Arrays with Objects
  - Command Line Arguments
  - Advantages & Disadvantages of Arrays
-

## Chapter 9: Inheritance & Polymorphism

- `extends` keyword
  - Single, Multilevel, Hierarchical Inheritance
  - `this` & `super` Keywords
  - `final` Variables, Methods, Classes
  - Abstract Classes & Methods
  - Interfaces, Implementation, Variables
  - Multiple Inheritance via Interfaces
  - Functional Interfaces & Lambda Expressions
  - Abstraction & Polymorphism
  - Method Overloading & Overriding
- 

## Chapter 10: Exception Handling & OOP Concepts

- Wrapper Classes, Autoboxing, Unboxing
  - Object Class & Methods
  - Type Casting & `instanceOf` Operator
  - Exception vs Error
  - Try-Catch-Finally Blocks
  - Checked & Unchecked Exceptions
  - `throws` Keyword
  - User-Defined Exceptions
  - Access Modifiers
  - Encapsulation, POJO, JavaBeans
- 

## Chapter 11: Collections, Threads & I/O

- Collections API (List, Set, Map)
  - Iterators & Stream API
  - Threads & Life Cycle
  - Java I/O Streams
  - Date & Time API
  - Inner Classes & Blocks
  - Garbage Collection
  - String Literal vs `new` Operator
-

## ADVANCED JAVA (J2EE)

### Chapter 1: JDBC API

- Introduction to Database & JDBC
- Installing MySQL/Oracle DB
- JDBC Drivers
- Connection, Statement, PreparedStatement
- Dynamic & Static SQL Queries
- ResultSet
- Closing Connections
- Transaction Management

### Chapter 2: Servlets API

- Web Server & Apache Tomcat Installation
- Creating Servlets & WAR Files
- Deployment Descriptor (web.xml)
- HTTP & HTTPS Protocols
- Request/Response Objects
- Generic & HTTP Servlets
- Servlet Lifecycle
- Context & Config
- Cookies & Sessions
- Session Management

## SPRING, SPRING BOOT & MICROSERVICES

### Chapter 3: Spring Framework

- Introduction to Frameworks
- IOC & Dependency Injection
- Spring Containers (BeanFactory, ApplicationContext)
- Bean Creation, Configuration, Autowiring
- MVC Architecture
- Spring Web MVC Flow
- Controllers, Services, Repositories
- CRUD Operations
- Transaction Management
- Security Basics

---

## Chapter 4: Spring Boot

- Spring Boot Overview & Features
- Annotations & Starters
- Dependency Management
- Profiles, Properties, Actuator, DevTools
- Exception Handling
- REST APIs, CRUD Examples
- Microservices & Architecture

---

## Chapter 5-8: Advanced Modules

- Core Module Implementation
- JDBC, ORM, JPA Integration
- Repository Interfaces, CRUD/JPA Repositories
- MVC & REST Controllers (@RestController, @RequestMapping, etc.)
- JSON, HTTP, Postman, Swagger
- Authentication (JWT, OAuth2)
- DevTools, Docker, Caching, React Integration
- Microservices Architecture, API Gateway, Circuit Breaker, Eureka, Config Server, Load Balancing

---

## FRONTEND DEVELOPMENT

### HTML

- HTML Basics, Tags, Structure
- Headings, Paragraphs, Lists, Tables, Forms
- Anchors, Images, Links, Multimedia (Audio, Video)
- Semantic Tags (header, footer, nav, section, etc.)
- Global Attributes & Linking Resources
- Embedding Content (SVG, IFrame)

---

### CSS

- Syntax, Selectors, Colors, Fonts, Box Model
- Margin, Padding, Borders, Backgrounds
- Flexbox, Grid, Positioning
- Animations, Transitions, Transformations
- Responsive Design (Media Queries)

---

## JAVASCRIPT

- Basics: Variables, Data Types, Operators
- Conditions, Loops, Functions
- Arrays, Objects, OOPs, Classes, Inheritance
- DOM Manipulation, Events
- Asynchronous JS (Promises, Async/Await)
- Error Handling, Web Storage, Regex
- Date, JSON, Local & Session Storage

---

## REACTJS

- JSX, Components, Props, State, Hooks
- useState, useEffect, useReducer
- Forms & Events Handling
- Routing (react-router-dom)
- Context API, Redux, Redux Toolkit
- Deployment & Bonus Topics (Next.js, TypeScript)

---

## SQL (MYSQL/MSSQL)

### Chapters Overview

1. **Introduction to Databases**
    - DBMS, SQL vs MySQL, Installation
  2. **Creating Tables**
    - Data Types, CREATE & DROP
  3. **Data Manipulation**
    - INSERT, UPDATE, DELETE, SELECT
  4. **Operators**
    - Logical & Comparison Operators
  5. **Functions**
    - SUM, AVG, COUNT, MIN, MAX, GROUP BY
  6. **Joins**
    - Inner, Left, Cross Joins
-

## Real-Time Project:

- Enterprise Resource Management System (ERMS): A full-stack web application integrating Spring Boot, ReactJS, and MySQL.
- Implements JWT authentication, role-based access, and RESTful APIs for modules like Employee, Project, and Finance Management.
- Follows Micro services Architecture using Spring Cloud, Eureka, and API Gateway.
- Includes complete end-to-end development with frontend, backend, and database integration.
- Covers the entire tech stack: Core Java, J2EE, Spring Boot, Micro services, ReactJS, SQL, and DevOps.

## PROJECT OUTCOMES

By completing the real-time project:

- You'll build and deploy a **complete full stack web application**.
- You'll understand **end-to-end development**, including frontend, backend, DB, and security.
- You'll gain confidence to work on **enterprise-level applications**.
- You'll have a **portfolio project** ready for interviews and placements.