# Java Full Stack Development Online Course

## Preface

Welcome to the **Java Full Stack Development Online Course** from a2v Academy!

In today's fast-paced digital world, the demand for skilled Full Stack Developers is higher than ever. Whether you're building dynamic web applications, enterprise-level software, or modern microservices, a deep understanding of both front-end and back-end development is essential.

This course is designed to guide you through the complete journey of becoming a proficient Full Stack Developer using Java — one of the most widely-used and trusted programming languages in the industry. From core Java programming to advanced frameworks like Spring Boot, from front-end technologies like HTML, CSS, and JavaScript to powerful databases and DevOps practices — this course brings it all together.

What makes this course unique is its focus on **practical, project-based learning**. By the end of this program, you won't just understand the theory — you'll have built **real-world applications** and developed the confidence to tackle complex development challenges.

Whether you are a beginner aiming to break into the tech industry or an experienced professional looking to upskill, this course provides the foundation, tools, and guidance you need to thrive in the world of full stack development.

## Course Outline

### Core Java

*This module provides a foundational understanding of the Java programming language.*

- **Object-Oriented Programming (OOP):** Master concepts like classes, objects, inheritance, encapsulation, polymorphism, and abstraction.
- **Data Structures and Collections:** Work with essential data structures such as ArrayList, LinkedList, HashMap, and HashSet etc.
- **Core Fundamentals:** Learn Java syntax, variables, data types, control flow statements (if-else, loops), arrays, methods, and exception handling.
- **Advanced Features:** Multi-threading, File I/O, Generics and streams.

## Front-end development

*This section provides an in-depth exploration of client-side frameworks and tools for constructing interactive and adaptive user interfaces.*

- **HTML5:** Structure the content of web pages using semantic tags, forms, and media elements.
- **CSS3:** Style web pages with properties for layout (Flexbox, Grid), typography, and responsive design using media queries.
- **JavaScript (ES6+):** Add dynamic and interactive behavior to websites. Topics include variables, data types, DOM manipulation, events, and asynchronous programming.
- **JavaScript Framework - React:** Use a modern framework like React to build component-based user interfaces. Learn about JSX, state management, props, and hooks.

## Back-end development (Server-side)

*This module focuses on implementing backend functionality and establishing communication with the front-end.*

- **Java Servlets and JSP:** Explore foundational Java technologies for creating dynamic web pages and handling HTTP requests and responses.
- **Spring Framework:** Learn the Spring ecosystem, including Spring Core for Dependency Injection (DI) and Inversion of Control (IoC), and Spring MVC for building web applications.
- **Spring Boot:** Use Spring Boot to simplify the development of production-ready, stand-alone, and microservice-based applications.
- **RESTful APIs:** Build and consume REST APIs using Spring Boot, focusing on HTTP methods (GET, POST, PUT, DELETE) and JSON data handling.
- **Hibernate and JPA:** Use Hibernate as an Object-Relational Mapping (ORM) tool to interact with databases by mapping Java objects to database tables.

## Database Management

*Get familiar with managing data using both relational and non-relational databases.*

- **Relational Databases (MySQL/PostgreSQL):** Learn database fundamentals, including schema design, normalization, and SQL queries (CRUD operations, joins).
- **NoSQL Databases (MongoDB):** Understand document-based storage and perform CRUD operations, querying, and indexing.
- **JDBC (Java Database Connectivity):** Use the standard Java API for connecting to and querying a database.

## DevOps and Deployment

*This segment highlights the tools and processes involved in automating the software release cycle.*

- **Version Control (Git/GitHub):** Learn to track changes in code, collaborate with other developers, and manage repositories.
- **Build Tools (Maven):** Use build automation tools to manage dependencies and build projects.
- **CI/CD (Jenkins):** Set up and use Continuous Integration and Continuous Deployment pipelines.
- **Containerization (Docker):** Package applications and their dependencies into portable containers for consistent deployment across environments.
- **Cloud Deployment:** Learn to deploy applications on cloud platforms like Amazon Web Services (AWS)

## Software Development Life Cycle using Agile

"Agile methodology is widely adopted by nearly all IT companies worldwide."

The Agile Software Development Life Cycle (SDLC) is an iterative and collaborative approach to building software that emphasizes flexibility, customer feedback, and rapid delivery.

- Introduction to SDLC and Agile
- Agile SDLC Phases
- Agile Planning and Design
- Agile Development Practices
- Agile Ceremonies, Roles and Tools

## Capstone Project

- Execute an end-to-end capstone project involving the architecture, development, and integration of front-end interfaces, back-end services, and database management systems to deliver a cohesive solution.
- The course culminates in an experiential learning project where you will build a fully functional application that applies all course concepts to solve a real-world problem

## Total Duration of the Course

- **Total duration:** 3 months.
- **Days:** Monday to Friday (1.5 hrs daily)
- **Mode:** Online

## Certificate

- A course completion certificate will be awarded once you have successfully completed this course.

Wishing you a **Smooth Coding** ahead!