

Full Stack Java Course

FA-0499 · 20-day

Software Development · DevOps, Cloud & Infrastructure · Cybersecurity · Data & Analytics

Introduction

This 20-day full stack Java course is designed to provide a comprehensive understanding of web development with a primary focus on Java and related technologies. Throughout the course, you will gain hands-on experience working with various tools and frameworks, including HTML, CSS, JavaScript, Core Java, Spring, Spring Boot, Hibernate, MS SQL, Microservices, VueJS, and deployment using DigitalOcean Droplet or AWS EC2 on linux and windows servers.

This course outline stands out from other training programs primarily due to the trainers' extensive industry experience, with each having over 20 years of individual expertise in the field. This wealth of practical knowledge allows them to design and deliver a comprehensive curriculum that is specifically tailored to meet the demands of today's rapidly evolving web development landscape. Their vast understanding of real-world projects enables them to incorporate the latest best practices, tools, and technologies, ensuring that participants receive a top-notch educational experience that goes beyond mere theoretical knowledge.

In Malaysia, a significant number of training programs fall short in providing coherent and deployable content for commercial or professional settings, often being more suitable for academic purposes. However, our course outline is meticulously crafted to cater to the actual needs of the industry, focusing on hands-on exercises, real-world examples, and the practical application of skills. By offering a perfect balance of theoretical concepts and practical implementation, this course equips participants with the necessary skills to excel in their professional careers, setting it apart from the majority of training programs available in Malaysia.

Learning outcomes

- Understand the Software Development Life Cycle (SDLC).
- Design and develop web applications using HTML, CSS, and JavaScript.
- Gain proficiency in Core Java and Object-Oriented Programming (OOP) concepts.
- Develop web applications using Spring, Spring Boot, and Hibernate.
- Implement Microservices architecture in Java applications.
- Integrate and utilize VueJS for front-end development.

- Deploy applications on DigitalOcean Droplet or AWS EC2.
- Understand and work with Ubuntu OS.

Prerequisites

- Basic knowledge of programming concepts.
- Familiarity with web development fundamentals.

Training outline

1. Day 1: Introduction

Course Introduction

- Welcome and introductions
- Course objectives and learning outcomes
- Overview of the course structure and topics

Introduction to the Software Development Life Cycle (SDLC) Overview of SDLC and its importance

- Stages of SDLC: Planning, Analysis, Design, Implementation, Testing, Deployment, and Maintenance
- Different SDLC models: Waterfall, Agile, Scrum, and others
- Role of full stack developers in SDLC

IDE Setup: IntelliJ

- Introduction to IntelliJ IDEA
- Downloading and installing IntelliJ IDEA
- Exploring the interface and features
- Creating a simple Java project
- Debugging and running a Java program in IntelliJ

IDE Setup: Visual Studio Code

- Introduction to Visual Studio Code
- Downloading and installing Visual Studio Code
- Exploring the interface and features
- Installing useful extensions for web development
- Creating a simple HTML, CSS, and JavaScript project
- Running a web application in Visual Studio Code

Recap and Q&A

- Review of the topics covered in Day 1

- Open discussion and questions
- Preview of the topics for Day 2

2. Day 2: Front-End Development Basics

Introduction to HTML

- Overview of HTML and its importance in web development
- HTML structure, elements, and attributes
- Commonly used HTML tags: headings, paragraphs, links, images, lists, and tables

Hands-on Exercise: Creating a Basic HTML Page

- Setting up a project in Visual Studio Code
- Creating an HTML file and adding common elements
- Previewing the page in a web browser

Introduction to CSS

- Overview of CSS and its importance in web development
- CSS syntax, selectors, and properties
- Inline, internal, and external CSS

:

Hands-on Exercise: Styling an HTML Page with CSS

- Adding CSS to the HTML project from the
- Styling the HTML elements using various CSS properties
- Previewing the styled page in a web browser

Introduction to JavaScript

- Overview of JavaScript and its importance in web development
- JavaScript syntax, variables, and data types
- Control structures: if, switch, for, and while

Hands-on Exercise: Adding Basic JavaScript Functionality

- Adding a JavaScript file to the HTML project
- Implementing simple JavaScript functions for user interactions
- Previewing the updated page with JavaScript functionality in a web browser

Recap and Q&A

- Review of the topics covered in Day 2
- Open discussion and questions
- Preview of the topics for Day 3

3. Day 3: Core Java and OOP Concepts

Java Basics

- Overview of Java and its importance in software development
- Java syntax, data types, and operators
- Control structures: if, switch, for, and while

Hands-on Exercise: Writing a Simple Java Program

- Setting up a Java project in IntelliJ IDEA
- Writing a simple Java program with control structures
- Compiling and running the Java program in IntelliJ IDEA

Object-Oriented Programming (OOP) Concepts in Java - Part 1

- Introduction to Object-Oriented Programming
- Classes, objects, and their relationship
- Constructors and method overloading
- Inheritance and the use of the 'extends' keyword

Hands-on Exercise: Creating Classes and Objects in Java

- Creating classes and objects in a Java project
- Implementing constructors and methods
- Creating a simple inheritance hierarchy

Object-Oriented Programming (OOP) Concepts in Java - Part 2

- Interfaces and the use of the 'implements' keyword
- Abstract classes and methods
- Polymorphism and method overriding
- Encapsulation: public, private, and protected access modifiers

Hands-on Exercise: Implementing Interfaces and Abstract Classes

- Creating interfaces and abstract classes in a Java project
- Implementing interfaces and extending abstract classes
- Demonstrating polymorphism and encapsulation

Recap and Q&A

- Review of the topics covered in Day 3
- Open discussion and questions
- Preview of the topics for Day 4

4. Day 4: Core Java - Advanced OOP and Exception Handling

Advanced Object-Oriented Programming in Java

- Nested classes: static and non-static
- Anonymous classes and lambda expressions
- The 'final' keyword: final variables, methods, and classes

Hands-on Exercise: Working with Advanced OOP Concepts

- Implementing nested classes and anonymous classes in a Java project
- Using lambda expressions for functional programming
- Applying the 'final' keyword in different scenarios

Java Exception Handling - Part 1

- Overview of exceptions in Java
- The try-catch-finally block
- Types of exceptions: checked and unchecked

Hands-on Exercise: Handling Exceptions in Java

- Writing Java code that generates exceptions
- Handling exceptions using try-catch-finally blocks
- Working with checked and unchecked exceptions

Java Exception Handling - Part 2

- Creating custom exceptions
- Best practices for exception handling
- The 'throws' keyword and exception propagation

Hands-on Exercise: Creating Custom Exceptions

- Defining and using custom exception classes
- Implementing exception propagation with the 'throws' keyword
- Applying best practices in exception handling

Recap and Q&A

- Review of the topics covered in Day 4
- Open discussion and questions
- Preview of the topics for Day 5

5. Day 5: Introduction to Spring Framework

Overview of Spring Framework

- Introduction to Spring Framework and its importance in Java development
- Core features of Spring Framework: Dependency Injection (DI) and Inversion of Control (IoC)
- Spring modules: Core, AOP, Data, Web, and more

Spring Framework - Basics

- Setting up a Spring project in IntelliJ IDEA

- Spring configuration using XML and annotations
- Creating beans and managing their lifecycle

Hands-on Exercise: Creating a Simple Spring Application

- Developing a Spring application using XML configuration
- Developing a Spring application using annotation-based configuration
- Testing the Spring application in IntelliJ IDEA

Dependency Injection (DI) and Inversion of Control (IoC)

- Understanding the concepts of Dependency Injection and Inversion of Control
- Types of Dependency Injection: constructor-based and setter-based
- Autowiring and its different modes: by type, by name, constructor, and auto-detect

Hands-on Exercise: Implementing Dependency Injection

- Creating beans with dependencies
- Implementing constructor-based and setter-based DI
- Exploring various autowiring modes

Aspect-Oriented Programming (AOP) with Spring

- Introduction to Aspect-Oriented Programming
- Spring AOP: aspects, pointcuts, advice, and join points
- Configuring AOP using XML and annotations

Hands-on Exercise: Implementing AOP in a Spring Application

- Creating aspects and pointcuts in a Spring application
- Defining different types of advice: before, after, around, after returning, and after throwing
- Testing AOP functionality in IntelliJ IDEA

Recap and Q&A

- Review of the topics covered in Day 5
- Open discussion and questions
- Preview of the topics for Day 6

6. Day 6: Spring Boot - Basics and Auto-Configuration

Introduction to Spring Boot

- Overview of Spring Boot and its advantages
- Spring Boot features: Auto-Configuration, Starters, and Embedded Web Server
- Comparing Spring Framework and Spring Boot

Setting Up a Spring Boot Project

- Creating a Spring Boot project using Spring Initializr
- Exploring the project structure and dependencies

- Running a Spring Boot application in IntelliJ IDEA

Spring Boot Auto-Configuration

- Understanding the Auto-Configuration process
- Analyzing the Auto-Configuration report
- Customizing Auto-Configuration using application.properties

Hands-on Exercise: Creating a Simple Spring Boot Application

- Developing a Spring Boot application with auto-configured components
- Customizing the auto-configuration settings
- Testing the Spring Boot application in IntelliJ IDEA

Spring Boot Starters

- Introduction to Spring Boot Starters
- Commonly used Starters: Web, JPA, Security, and more
- Creating custom Spring Boot Starters

Hands-on Exercise: Working with Spring Boot Starters

- Adding Starters to a Spring Boot project
- Configuring and using various Spring Boot Starters
- Testing the Spring Boot application with integrated Starters

Recap and Q&A

- Review of the topics covered in Day 6
- Open discussion and questions
- Preview of the topics for Day 7

7. Day 7: Spring Boot - Web Development and Data Access

Spring Boot Web Development

- Introduction to Spring Boot Web development
- Spring MVC architecture and components
- Creating controllers, views, and templates using Thymeleaf

Hands-on Exercise: Developing a Spring Boot Web Application

- Setting up a Spring Boot project with the Web Starter
- Creating a simple web application with controllers and views
- Implementing navigation and user interactions

Spring Boot Data Access

- Overview of data access in Spring Boot
- Introduction to Spring Data JPA
- Repositories, entities, and CRUD operations

Hands-on Exercise: Implementing Data Access in a Spring Boot Application

- Configuring a Spring Boot project with Spring Data JPA
- Creating entities and repositories
- Implementing CRUD operations in the web application

Spring Boot Database Integration

- Integrating Spring Boot with various databases
- Configuring data sources and connection properties
- Using embedded and external databases

Hands-on Exercise: Integrating a Database with Spring Boot

- Connecting the Spring Boot application to a database
- Configuring the data source and connection properties
- Testing database connectivity and CRUD operations

Recap and Q&A

- Review of the topics covered in Day 7
- Open discussion and questions
- Preview of the topics for Day 8

8. Day 8: Spring Boot - Security and Testing

Introduction to Spring Boot Security

- Overview of Spring Boot Security and its importance
- Authentication and authorization concepts
- Configuring Spring Boot Security with default settings

Hands-on Exercise: Implementing Basic Spring Boot Security

- Setting up a Spring Boot project with Spring Security Starter
- Configuring default security settings
- Testing authentication and authorization in the web application

Spring Boot Security Customization

- Customizing authentication and authorization settings
- Creating and configuring custom user roles and permissions
- Implementing password encryption and user management

Hands-on Exercise: Customizing Spring Boot Security

- Customizing the security settings in a Spring Boot application
- Implementing custom user roles and permissions
- Testing the updated security settings in the web application

Spring Boot Testing

- Overview of testing in Spring Boot
- Unit testing with JUnit and Mockito
- Integration testing with Spring Boot Test

Hands-on Exercise: Testing a Spring Boot Application

- Writing unit tests for controllers, services, and repositories
- Writing integration tests for the web application
- Running tests and analyzing results in IntelliJ IDEA

Recap and Q&A

- Review of the topics covered in Day 8
- Open discussion and questions
- Preview of the topics for Day 9

9. Day 9: Hibernate - Basics and Configuration

Introduction to Hibernate

- Overview of Hibernate and its importance in Java development
- Object-Relational Mapping (ORM) and its benefits
- Hibernate architecture and components

Hibernate Configuration

- Setting up a Hibernate project in IntelliJ IDEA
- Configuring Hibernate with XML and annotations
- Creating a SessionFactory and managing sessions

Hands-on Exercise: Creating a Simple Hibernate Application

- Developing a Hibernate application with XML configuration
- Developing a Hibernate application with annotation-based configuration
- Testing the Hibernate application in IntelliJ IDEA

Hibernate Entities and Mappings

- Creating and configuring entity classes
- Mapping relationships: one-to-one, one-to-many, and many-to-many
- Fetch strategies and lazy loading

Hands-on Exercise: Implementing Hibernate Entities and Mappings

- Creating entity classes for a Hibernate project
- Implementing various relationship mappings
- Testing the entity mappings in the Hibernate application

CRUD Operations with Hibernate

- Hibernate Query Language (HQL) and Criteria API

- Inserting, updating, deleting, and querying data with Hibernate
- Transactions and concurrency management

Hands-on Exercise: Implementing CRUD Operations in a Hibernate Application

- Writing HQL queries and using the Criteria API
- Performing CRUD operations in the Hibernate application
- Testing the CRUD functionality in IntelliJ IDEA

Recap and Q&A

- Review of the topics covered in Day 9
- Open discussion and questions
- Preview of the topics for Day 10

10. Day 10: Integrating Hibernate with Spring Boot and Introduction to Microservices

Spring Boot and Hibernate Integration

- Overview of Spring Boot and Hibernate integration
- Configuring Spring Boot to use Hibernate
- Creating a Spring Boot project with Spring Data JPA and Hibernate

Hands-on Exercise: Integrating Hibernate into a Spring Boot Application

- Setting up a Spring Boot project with Hibernate
- Implementing entities, repositories, and services
- Testing the integrated Spring Boot and Hibernate application

Introduction to Microservices

- Overview of microservices architecture
- Benefits and challenges of microservices
- Microservices communication patterns and best practices

Spring Boot Microservices

- Introduction to Spring Boot Microservices
- Creating and configuring microservices using Spring Boot
- Service discovery, load balancing, and API Gateway with Spring Cloud

Hands-on Exercise: Creating Spring Boot Microservices

- Developing a simple microservices-based application using Spring Boot
- Implementing service discovery and load balancing with Spring Cloud
- Testing the microservices application in IntelliJ IDEA

Recap and Q&A

- Review of the topics covered in Day 10

- Open discussion and questions
- Preview of the topics for Day 11

11. Day 11: Vue.js - Basics and Project Setup

Introduction to Vue.js

- Overview of Vue.js and its importance in modern web development
- Comparison with other JavaScript frameworks (React, Angular)
- Core concepts: components, directives, and reactive data

Vue.js Project Setup

- Installing Node.js and the Vue CLI
- Creating a Vue.js project using the Vue CLI
- Exploring the project structure and components

Hands-on Exercise: Creating a Simple Vue.js Application

- Developing a Vue.js application with components, directives, and reactive data
- Implementing navigation and user interactions
- Testing the Vue.js application in the browser

Vue.js Components

- Understanding the component-based architecture
- Creating and using components in a Vue.js application
- Component communication: props and custom events

Hands-on Exercise: Working with Vue.js Components

- Developing a Vue.js application with multiple components
- Implementing component communication using props and custom events
- Testing the component interactions in the browser

Vue.js Directives and Filters

- Overview of built-in Vue.js directives: v-if, v-for, v-bind, v-model, and more
- Creating custom directives
- Using filters for data formatting

Hands-on Exercise: Implementing Directives and Filters in a Vue.js Application

- Applying built-in directives in a Vue.js application
- Developing custom directives and filters
- Testing the updated Vue.js application in the browser

Recap and Q&A

- Review of the topics covered in Day 11
- Open discussion and questions

- Preview of the topics for Day 12

Days 12 - 13: Vue.js - State Management and Integration with Spring Boot

Introduction to Vuex

- Overview of Vuex and its importance in state management
- Core concepts: state, getters, mutations, actions, and modules
- Integrating Vuex with a Vue.js application

Hands-on Exercise: Implementing Vuex in a Vue.js Application

- Setting up Vuex in a Vue.js project
- Developing a Vuex store with state, getters, mutations, and actions
- Implementing state management in the Vue.js application

•

Vue.js HTTP Requests

- Overview of HTTP requests in Vue.js
- Using Axios for making HTTP requests
- Integrating Axios with a Vue.js application

• :

Hands-on Exercise: Implementing HTTP Requests in a Vue.js Application

- Configuring Axios in a Vue.js project
- Developing components that make HTTP requests to a backend API
- Testing the HTTP requests in the browser

Integrating Vue.js with Spring Boot

- Overview of integrating Vue.js with Spring Boot
- Creating a Spring Boot backend API for the Vue.js application
- Configuring CORS and security settings in Spring Boot

Hands-on Exercise: Connecting Vue.js to a Spring Boot Backend

- Developing a Spring Boot API for the Vue.js application
- Connecting the Vue.js application to the Spring Boot backend
- Testing the integrated application in the browser

Recap and Q&A

- Review of the topics covered in Day 12
- Open discussion and questions
- Preview of the topics for Day 13

Days 14 - 16: Deployment

Introduction to Ubuntu

- Overview of Ubuntu and its importance in web development
- Basic Ubuntu commands and directory structure
- Installing and managing software packages

DigitalOcean Droplets

- Overview of DigitalOcean and its features
- Creating and managing Droplets
- Configuring domain names and SSL certificates

AWS EC2

- Overview of Amazon Web Services (AWS) and its features
- Introduction to EC2 instances
- Creating and managing EC2 instances, security groups, and key pairs

Deploying a Spring Boot Application to DigitalOcean

- Preparing a Spring Boot application for deployment
- Creating a DigitalOcean Droplet with Ubuntu
- Deploying the Spring Boot application to the Droplet

Deploying a Spring Boot Application to AWS EC2

- Preparing a Spring Boot application for deployment
- Creating an AWS EC2 instance with Ubuntu
- Deploying the Spring Boot application to the EC2 instance

Hands-on Exercise: Deploying a Spring Boot and Vue.js Application

- Preparing a full-stack application (Spring Boot and Vue.js) for deployment
- Choosing a deployment platform: DigitalOcean or AWS EC2
- Deploying the application and testing it on the chosen platform

Introduction to Docker

- Overview of Docker and its importance in modern web development
- Docker architecture: images, containers, and registries
- Basic Docker commands and Dockerfile

Hands-on Exercise: Dockerizing a Spring Boot Application

- Creating a Dockerfile for a Spring Boot application
- Building and running a Docker container
- Testing the Dockerized Spring Boot application

Introduction to Kubernetes

- Overview of Kubernetes and its role in container orchestration

- Kubernetes architecture and components: nodes, pods, services, and deployments
- Basic Kubernetes commands and configuration files

Hands-on Exercise: Deploying a Spring Boot Application to Kubernetes

- Creating a Kubernetes deployment configuration for a Spring Boot application
- Deploying the application to a Kubernetes cluster
- Scaling and managing the application in Kubernetes

Introduction to Continuous Integration/Continuous Deployment (CI/CD)

- Overview of CI/CD and its importance in web development
- Popular CI/CD tools and platforms: Jenkins, Travis CI, CircleCI, and GitHub Actions
- Basic CI/CD concepts: pipelines, stages, and jobs

Hands-on Exercise: Implementing a CI/CD Pipeline

- Choosing a CI/CD platform and setting up an account
- Configuring a CI/CD pipeline for a Spring Boot application
- Deploying the application automatically through the pipeline

Recap and Q&A

- Review of the topics covered in Day 14
- Open discussion and questions
- Preview of the topics for Day 15

Days 17 - 22: Capstone

Java Concurrency

- Overview of Java concurrency and multithreading
- Creating and managing threads in Java
- Synchronization and thread safety

Hands-on Exercise: Implementing Concurrency in Java

- Developing a Java application with multithreading
- Implementing synchronization and ensuring thread safety
- Testing the concurrent functionality in IntelliJ IDEA

Java Performance Optimization

- Overview of Java performance optimization
- Common performance issues and bottlenecks in Java applications
- Profiling and monitoring tools for Java

Final Project Overview

- Introduction to the final project and its requirements
- Reviewing the project's objectives and milestones

- Forming teams and assigning roles

Final Project Work Session

- Teams work on their respective projects, applying the skills learned throughout the course
- Instructor and teaching assistants provide guidance and support

Final Project Presentations

- Each team presents their project to the class, demonstrating their application's features and functionality
- Instructor and teaching assistants provide feedback and suggestions

Course Recap and Review

- Review of the main topics covered during the course
- Discussion of best practices and industry standards
- Open Q&A session

Exploring Next Steps in Web Development

- Recommendations for further learning and skill development
- Introduction to related technologies and frameworks
- Networking and job search strategies in the web development field

Course Evaluation and Feedback

- Participants complete course evaluations
- Discussion of feedback and suggestions for future course improvements