



Masters in Full Stack Development Training Curriculum

STRUCTURE



Masters in Full Stack Development Training Curriculum

About Croma Campus:

Croma Campus Training & Development Private Limited is an education platform since 2010 providing rigorous industry-relevant programs designed and delivered in collaboration with world-class faculty and industry.

- Hands-On Live Projects
- Simulation Test Papers
- Industry Cases Studies
- 61,640+ Satisfied Learners
- 140+ Training Courses
- 100% Certification Passing Rate
- Live Instructor Classroom / Online Training
- 100% Placement Assistance

Course Objectives:

- Learn core computer science concepts from leading industry experts with content structured to ensure industry relevance
- Build an end-to-end application with exciting features and test it
- Earn an industry-recognized course completion certificate
- Make you career-ready as a full stack developer upon successful completion of the program.

Croma Campus Training Program Deliverables:

- **Session Recordings** - Original Class Room Voice & Video Recording
- **Training Material** - Soft Copy Handbooks
- **Assignments** | Multiple Hands-on Exercises
- **Test Papers** - We provide **Practice Test** as part of our course to help you prepare for the actual certification exam.
- **Live Case Studies**
- **Live Projects** - Hands-on exercises and Project work. You will work on real time industry-oriented projects and assignments for each module to practice.
- **Key focus on Hands-on exercises and Project work.** You will work on real time industry-oriented projects.
- Faculty with more than **10+ Years of Experience** in the Industry.
- **Technical Resume Designing & Job Assistance:** With more than 100+ Clients across the Globe and we help learners to get a good job in their respective field. We also help learners with resume preparation.
- **Interview Q&A**
- **About Croma Campus Training Certificate:** Croma Campus will provide you with an industry-recognized (Certified by **ISO 9001:2015 & E-Cell IIT Jodhpur**) course completion certificate which has lifelong validity.
- **How I Unlock my Croma Campus Certificate:** Attend Complete Batch & Submit at least One Completed Project.

Master in Full Stack Development Course Content

Course 1: Core Java Training

Java Programming language is one of the most used and most popular languages in the world of IT development. Given its properties of platform independence, object-oriented approach, reusable code and dynamic classes – Java has emerged to be the most preferred language for programmers.

Our Java programming courses cover the length and breadth of this platform-independent and object-oriented language that one can learn and implement easily for programming purposes. As the language enables faster coding with fewer bugs, it is the favourite of programmers. Our training course offers the expertise of Java to all who want to use it for coding on different devices.

Module 1: Java Programming Basics

- Introduction to Java
- Bytecode
- Class Files
- Compilation Process
- JDK 10 Installation
- Eclipse IDE Installation
- Writing First Java Program
- Primitive Data Types in Java
- Arithmetic Operators in Java
- Logical and Bitwise Operators
- Relational Operators in Java
- If - Condition
- Nested If - Condition
- For Loop in Java
- Nested “For” Loop
- 'While' & 'Do While' Loop in Java
- Loop 'Break' & 'Continue' Statements
- What's new in Java 10
- 'var' Keyword
- Concept of Garbage Collector (GC)
- Java 10 Update in GC
- Other Updates in Java 10

Module 2: Object-Oriented Programming in Java

- Concept of Object Orientation
- Classes and Objects
- Attributes, and Methods
- Method Overloading

- Method Overriding
- Access Specifiers in Java
- Constructors in Java
- Default constructors
- Constructors with arguments
- Data Encapsulation
- Static Keyword
- Concept of Main Method
- Inheritance
- Polymorphism
- Super Keyword
- Super Class Constructor
- Protected Access
- Abstraction
- Interface
- Final Keyword

Module 3: Data Handling and Functions

- Arrays
 - Array Objects
 - Single Dimensional Arrays
 - Multidimensional arrays
- Functions
 - Function with Arguments
 - Function Overloading
 - The concept of Static Polymorphism
- String Handling
 - String Basics
 - String Comparison Operations
 - String Search Operations
 - String (Cut) Slice Operations
 - String Replace Operations
 - String Conversion Operations
- Special Classes in Java
 - Math Class
 - StringBuilder Class
 - StringBuilder Methods
 - Scanner Class
 - Random Class
 - UUID Class
 - String Buffer Classes

Module 4: Packages, Multithreading, Exceptional Handling in Java

- Packages and Interfaces
- Access Specifiers: Public, Private, Protected and Package

- Exception Handling: Try, Catch, Finally, Throw and Throws
- What is an Exception?
- Error Vs Exception
- Checked and Unchecked Exception
- Throws Declaration
- Try and Catch Block (Exception Handling)
- 'Finally' Block
- Multi-Threading
- Runnable Interface
- Extending a Thread Class
- Synchronization in Threads

Module 5: Data structure Concepts in Java

- Lists
 - Array List
 - Linked List Structure
 - Linked List
 - ArrayList vs LinkedList
 - List Iterator
- Sets
 - Hash Set
 - Linked Hash Set
 - Tree Set
 - Iterating over Sets
- Maps
 - Hash Map
 - Tree Map
 - Iterating on Maps
- Regular Expressions
 - Introduction to Reg Ex
 - Quantifiers in Reg Ex
 - Character Classes in Reg Ex
 - Bracket Expressions in Reg Ex
 - OR Operator
 - Dot Operator
 - Greedy and Lazy Matching
 - Regularizing Number Ranges

Module 6: Data and Time Operations in Java

- Local Date and Time
- Custom Date and Time
- Future and Past Date
- Future and Past Time
- Date Difference
- Time Difference

- Date Time Formatter

Module 7: Working with Text files and Excel Files in Java

- Text Files Handling
 - Read Text File using Apache Commons IO
 - Edit Text File using Apache Commons IO
 - Sort Line from Text File
 - Copy & Move Text Files
 - Remove Duplicate Words from Text File
- Excel Files in Java/Apache POI
 - Apache POI Setup
 - Read Excel using POI
 - Read Excel Data into 2D Array
 - Write Excel using POI

Module 8: Java Collections

- Wrapper Classes and Inner Classes
 - Integer,
 - Character,
 - Boolean,
 - Float, etc
- Applet Programs
- How to write UI programs with Applet, Java.lang, Java.io, Java.util?
- Collections
 - ArrayList,
 - Vector,
 - HashSet,
 - TreeSet,
 - HashMap,
 - HashTable, etc.

Module 9: XML in Java

- Introduction to XML
- Writing XML files
- DOM Parser
- Writing into an XML file
- Parsing an XML file
- SAX Parser
- XSL

Module 10: JDBC

- Introduction to SQL
- SQL Commands
- Connect, Insert, Update, Delete, Select
- Introduction to JDBC and Architecture of JDBC

- Types of Drivers: Type 1/2/3/4 drivers
- Insert/Update/Delete/Select Operations using JDBC
- Batch Processing Transaction
- Management: Commit and Rollback

Module 11: Servlets

- Introduction to Web Technologies
- Type of Servlets
- Generic Servlet
- Http Servlet
- Request Dispatchers
- Forward and Include 4 types of Session
- Tracking and Filters

Module 12: Java Server Page / JSP

- Introduction to JSP
- Architecture of JSP
- tags (Scripts, declarative, expression)
- Implicit objects
- JSP Directives
- JSP and JDBC

Module 13: Hibernate

- Introduction to Hibernate
- Architecture of Hibernate
- Database Operations: Insert/Update/Delete/Select
- Inheritance
- Collections
- HQL and Restrictions
- Caching in Hibernate

Module 14: Springs in Java

- Introduction to Spring Framework
- Architecture
- Display a Sample Message
- IoC Containers
- Bean Definition
- Bean Scopes
- Bean Post Processors
- Dependency Injection Auto-Wiring

Module 15: AJAX and Design Patterns

- Aspect Oriented Programming (AOP)

- Integrating Spring framework with Hibernate
- Transaction Management
- Ajax Framework and Design Patterns: DAO, DTO, MVC
- Intercepting filters
- Front Controller
- Business Delegate

Module 16: SOA

- Introduction to SOA
- SOA Architecture
- Business layer of SOA
- Advantages of SOA
- What is Contract
- Address, and Binding in SOA
- Composition of Service
- Relation between SOA and Web Services

Module 17: Web Services and Projects

- Introduction to Web Services
- WSDL file
- WSDL and UDDI
- SOAP, RESTful Web Service
- JAX-WS Implementation
- Java Projects

Module 18: Placement Guide

- Tips to clear an Interview
- Common Interview questions and answers
- Core Java Interview Questions and Answers
- Resume Building Guide
- Attempt for Core Java Global Certification Exam
- Start applying for Jobs

Course 2: Angular Training

Mastering Angular is a must for developers who want to make a mark in the field of app, desktop and mobile development. It's an extremely popular JavaScript framework that has been widely adopted by organisations because of the unparalleled user experience it provides.

Our Angular Training Course at Croma Campus is designed as per the Industrial trends with 100% Job Assistance. We have a team of highly experienced professionals' faculties with more than 12 years of experience in Angular and other related Technologies.

Module 1: Angular Overview

- History of Angular
- The leap from AngularJS to Angular
- What's new in Angular 10
- Angular 10 vs Angular 9
- Desktop Application class User Experience
- Productivity and Tooling
- Performance
- Community
- Full-featured Framework
- Supported Browsers (Angular 10)
- Platform for Targeting Native Mobile not just Web Browsers

Module 2: All about TypeScript

- Introduction
- What is Typescript?
- Why Typescript?
- Setup and installation
- IDE support
- Different typescript versions
- Typescripts 3.8 for Angular 10
- Scoping using let and const Keywords (ES6)
- Template Literals (ES6)
- Rest and Spread Parameters (ES6)
- De-structuring (ES6)
- Introduction to Types
- Type inference
- Type Annotations
- Number
- Boolean
- String
- Array
- Tuple
- ENUM
- Any
- Void
- Null and Undefined
- Never
- Introduction to Functions
- Using types in functions
- Function as types
- Optional and default parameters
- Arrow functions
- Introduction to Classes
- Inheritance

- Access modifiers
- Getters and setters
- Read-only & static
- Introduction to Interfaces
- Optional properties and methods
- Strict structural contract
- Extending interface
- Implementing interface
- Introduction to Modules
- Import / Export
- Default
- Decorators

Module 3: Angular CLI

- Angular CLI
- Anatomy of the project
- Setting up a workspace
- Updating Angular apps using ng update
- Adding support for external libraries using ng add
- Debugging Angular apps
- Working with Augury
- Using the Angular Language Service with Microsoft VS Code

Module 4: Angular building blocks

- Modules
- Components
- Templates
- Metadata
- Data binding
- Directives
- Services
- Dependency injection
- Angular Ivy

Module 5: Angular Modules

- Why modules?
- How to create modules?
- Built-in modules
- Root Module
- feature module

Module 6: Components in Angular

- Introduction

- @Component decorator
- Component configuration object
- Custom components
- Component with templates
- Inline
- External
- Component with Styles
- Inline
- External
- Angular Elements

Module 7: Templates in Angular

- HTML basic syntax
- Template expressions
- Template syntax
- Attribute, class, and style bindings
- @Input ()
- @Output
- Template reference variables
- Safe navigation operator

Module 8: Data binding

- Interpolation
- Event binding
- Property binding
- two-way binding
- uses and examples

Module 9: Angular Directive and Pipes

- Introduction to Directives
- Built in Structural Directives
- Built in Attribute Directives
- Building Custom Attribute Directives
- Building Custom Structural Directives
- Introduction to Pipes in Angular
- Using Built-in Pipes
- Creating Custom Pipes in Angular
- Pure and Impure Pipes in Angular

Module 10: Angular Forms

- Types of Form in Angular
- Reactive form in Angular
- Dynamically Adding or Removing Form Control (s) or Form Group (s) using Form Array(s)

- Adding Synchronous Custom Validations to your Reactive Form
- Adding Asynchronous Custom Validations to your Reactive Form
- Resetting the value of a form

Module 11: Dependency Injections

- Introduction
- Why DI?
- @Injectable decorator
- Custom service development
- Registering the service with Ng Module using providers key
- Provider Types
- Class
- Factory
- Value

Module 12: Angular Routing with Guards

- Implementing Routing in an Angular App
- Abstracting the user link to a separate component
- Implementing Child Routes
- Path Match and Route Types
- Introduction to Route Guards in Angular
- CanActivate and CanActivateChild Guards in Angular
- CanDeactivate Guard in Angular
- Prefetching Data for a Component using Resolve

Module 13: RxJs Primer

- Introduction
- Why RxJs?
- Observable interface
- Streams
- Operators
- Subscription
- Subject
- Schedulers

Module 14: Back-end services, HTTP Client

- Setup installing the module
- Making a request for JSON data
- Type checking the response
- Error handling
- Sending data to the server
- Making a POST request
- Configuring other parts of the request

Module 15: Angular Advanced Features

- Creating Libraries
- Angular Material Essentials
- Lazy Loading
- Customizing the Angular CLI using the Builder API
- Server Side Rendering with Angular Universal
- Working with Service Workers
- Building a Progressive Web App

Module 16: Testing

- Creating Libraries
- Angular Material Essentials
- Lazy Loading
- Customizing the Angular CLI using the Builder API
- Server Side Rendering with Angular Universal
- Working with Service Workers
- Building a Progressive Web App

Module 17: Learn Deployment

- Manually
- Using the Angular CLI with Ahead-Of-Time (AOT) Compilation and Tree-Shaking (removing unused library code)
- Deployment Platforms for Angular Apps

Course 3: React JS Training

ReactJS presents graceful solutions to some of front-end programming's most persistent issues. It's fast, scalable, flexible, powerful, and has a robust developer community that's rapidly growing. React JS is a new age UI (user interface) tool that is more interactive and flexible as compared to most other tools. It comprises of components which are re-usable and easy to use.

Our React JS Training Course will help you master the fundamentals of React—an important web framework for developing user interfaces—including JSX, props, state, and events. This training course dives into Redux, covering topics like reducers, actions, and the state tree.

You will learn how to use ReactJS and the Redux library to create next gen web applications. It covers all the practical aspects of developing with React, managing data and server communication with Redux.

Module 1: Introduction of React.JS

- Audience
- Pre-requisites
- About React
- Features

- Advantages
- Limitations

Module 2: Environment Setup

- Create of Root Folder
- Install Global Packages
- Add Dependencies and Plugins
- Create the Files
- Set Compiler, Server and Loaders
- html
- JSX and Main.js
- Running the Server

Module 3: JSX

- What is JSX?
- Using JSX
- Nested Elements
- Attributes
- JavaScript Expressions
- Styling
- Components
- Naming Convention

Module 4: Components

- Stateless
- State full

Module 5: State

- What is State?
- Props

Module 6: Props Overview

- Using Props
- Default Props
- State and Props

Module 7: Props Validation

- Validating Props

Module 8: Component API

- Set State
- Force Update
- Find DOM Node

Module 9: Component Life Cycle

- Lifecycle Methods

Module 10: Forms

- Simple
- Complex

Module 11: Events

- Simple
- Child

Module 12: Refs

- What is Refs?
- Using Refs

Module 13: Keys

- What are Keys?
- Using Keys

Module 14: Router

- Install a React Router
- Add a Router
- Create Components

Module 15: Flux Concept

- What is Flux?
- Flux Elements
- Flux Props

Module 16: Using Flux and REDUX

- Install REDUX
- Creating Components
- Working with States
- Composing Components
- Actions
- Event Handlers
- Reducers
- Store
- Root Component
- Other Components
- Unit Testing – Tools, React, REDUX

Module 17: Animations

- Install React CSS Transitions Group
- Add a CSS File
- Appear Animation
- Enter and Leave Animations

Module 18: Higher-Order Components

- What is Higher-Order Component?

Course 4: Node JS Training

Node.js is an Open Source server framework, completely free, and used by thousands of developers around the world. Node.js allows you to run JavaScript on the server. Node.js is a very powerful JavaScript-based framework/platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications.

Our NodeJS Course at Croma Campus is designed as per the Industrial trends with 100% Job Assistance. We have a team of highly experienced professionals' faculties with more than 12 years of experience in Node JS and other related Technologies.

Module 1: Introduction

- Audience
- Pre-requisites
- About Node
- Execute Node
- Features
- Who use Node
- Concepts
- Where to use
- Where not to use

Module 2: Environment Setup

- Text Editor
- NodeJs Run Time
- Download NodeJs
- Installation
- Executing

Module 3: First Application

- Creating a NodeJs Application
- Make a request to NodeJs Server

Module 4: REPL Terminal

- What is REPL?
- Starting REPL
- REPL Commands
- Stopping REPL

Module 5: Packager Manager (NPM)

- Installing Modules using NPM
- Global vs Local Installation
- Using packages.json
- Attributes of packages.json
- Uninstalling Module
- Updating Module
- Searching Module
- Create a Module

Module 6: Call backs Concept

- Using Props
- Default Props
- State and Props

Module 7: Event Loop

- Validating Props

Module 8: Event Emitter

- Set State
- Force Update
- Find DOM Node

Module 9: Buffers

- Lifecycle Methods

Module 10: Streams

- Simple
- Complex

Module 11: File System

- Simple
- Child

Module 12: Global Objects

- What is Refs
- Using Refs

Module 13: Utility Modules

- What is Keys
- Using Keys

Module 14: Web Modules

- Install a React Router
- Add a Router
- Create Components

Module 15: Express Framework

- What is Flux
- Flux Elements
- Flux Props

Module 16: Restful API

- Install REDUX
- Create Files and Folders
- Actions
- Reducers
- Store
- Root Component
- Other Components

Module 17: Scaling Application

- Install React CSS Transitions Group
- Add a CSS File
- Appear Animation
- Enter and Leave Animations

Module 18: MongoDB

- Connecting Node and MongoDB
- Database Creation, Drop
- Collection Operations
- Documents Operations

Module 19: Application

- Node and MongoDB Application

Course 5: Mongo DB Training

MongoDB is an extremely helpful NoSQL database that is utilised by some of the world's largest companies. Our MongoDB certification training course will provide you with the necessary abilities to work with NoSQL databases in the best firms.

Module 1: MongoDB Basics

- Introduction, Advantages
- History, Features
- No SQL Databases
- Advantages over RDBMS

Module 2: Environment Setup

- Install MongoDB
- MongoDB Shell
- MongoDB Data Model
- MongoDB Datatypes

Module 3: Database

- Create Database
- Drop Database

Module 4: Collection

- Create Collection
- Drop Collection

Module 5: CRUD Documents

- Insert Documents
- Update Documents
- Delete Documents
- Query Documents

Module 6: Methods

- Limit ()
- Sort ()
- Skip ()

Course 6: Spring Boot Training

Spring Boot is a powerful framework, used to build web applications quickly with less code. The Course will cover how to use Spring Boot to build the various projects with knowledge. The content of the course is designed carefully by industry experts keeping latest Java trends in mind. We tried to cover everything that can be included under Spring Boot Category to make you more competent and knowledgeable resource in the end.

Module 1: Introduction to Web Services

- What are Web Services?
- Why do you need them?
- Different Types of Web Services
- Web Services Key Terminologies
- Introduction to SOAP Web Services
- Introduction to Restful Web Services
- SOAP vs Restful Web Services

Module 2: Spring Framework

- Introduction to Spring Framework
- Setting up a Spring Project
- Understanding Tight Coupling using the Binary Search Algorithm Example
- Making the Binary Search Algorithm Example Loosely Coupled
- Using Spring to Manage Dependencies - @Component, @Autowired
- What is happening in the background?
- Dynamic auto wiring and Troubleshooting - @Primary
- Constructor and Setter Injection
- Spring Modules
- Spring Projects
- Why is Spring Popular?

Module 3: Spring Boot

- Types of software architectures
- SOA and Monolith Architecture
- Why Microservices?
- Detailed Micro Services Architecture
 - App Layer
 - Business Layer
- Enterprise Layer
 - Infra Layer
- Advantages with Micro Services
- Introduction to Spring Boot
- Goals and Important features
- Developing Spring apps before Spring Boot
- Normal Spring Manual Approach

- Maven Overview
- Using Spring Initializer
- STS
- Eclipse with STS Plugin
- Create a simple rest controller
- What is Spring Boot Auto Configuration?
- Spring vs Spring Booth vs Spring MVC
- Spring Boot Starter Projects - Starter Web and Starter JPA
- Overview of different Spring Boot Starter Projects
- Spring Boot Actuator
 - Production Monitoring
 - Health Check Concepts
 - Security Measurements
- Spring Boot Developer Tools
- Spring Boot Custom Logging
 - Logging Level
 - Patterns Changes
 - Rolling Logs
- Spring Boot Profile Components
 - Introduction
 - Multiple Properties
 - YML File
 - Command Line Runner Example
 - Real time scenarios of components
- Auto Configuration
 - Introduction
 - @Conditional Flow
 - Customize conditional annotations
 - Spring Boot built in conditional annotations

Module 4: SOAP Web Services with Spring & Spring Boot

- SOAP Web Services Overview
- Initialize a Spring Web Services application with Spring Boot
- Overview of creating SOAP Web Service using Contract First Approach
- Define Request and Response XML Structure
- Define XML Schema Definition (XSD) for Request
- Define XML Schema Definition (XSD) for Response
- More about XML Schema Definition and Implementing XSD Best Practices
- Introduction to Java API for XML Binding (JAXB) and Configuring JAXB 2
- Configuring an Endpoint for GetCourseDetailsRequest
- Spring Web Services Configuration - Message Dispatcher Servlet
- Spring Web Services Configuration - Generating WSDL
- Using Wizzler to execute SOAP Requests
- Implementing a service - Course Details Service - backend with in memo

- Implementing SOAP Web Service for GetAllCourseDetailsRequest
- Quick introduction to different parts of a WSDL
- Implementing SOAP Web Service for DeleteCourseDetailsRequest
- Improving the DeleteCourseDetailsRequest - Using an Enum for Status
- Exception Handling and SOAP Fault Responses
- Implementing Security for SOAP Web Services with WS Security

Module 5: Restful Web Services with Spring & Spring Boot

- Introduction to Restful Services
- Initializing a RESTful Services Project with Spring Boot
- Understanding the RESTful Services
- Creating a Hello World Service
- Enhancing the Hello World Service to return a Bean
- Quick Review of Spring Boot Auto Configuration and Dispatcher Servlet
- Enhancing the Hello World Service with a Path Variable
- Creating User Bean and User Service
- Implementing GET Methods for User Resource
- Implementing POST Method to create User Resource
- CODE BACKUP FILE: For Reference
- Enhancing POST Method to return correct HTTP Status Code and Location
- Implementing Exception Handling - 404 Resource Not Found
- Implementing Generic Exception Handling for all Resources
- User Post Resource and Exception Handling
- Implementing DELETE Method to delete a User Resource
- Add dependency spring-boot-starter-validation
- Implementing Validations for RESTful Services
- HATEOAS Updates
- Implementing HATEOAS for RESTful Services
- Overview of Advanced RESTful Service Features
- Internationalization for RESTful Services
- Internationalization
- Content Negotiation - Implementing Support for XML
- Disable XML Format Support
- Configuring Auto Generation of Swagger Documentation
- Introduction to Swagger Documentation Format
- Enhancing Swagger Documentation with Custom Annotations
- Monitoring APIs with Spring Boot Actuator
- Implementing Static Filtering for RESTful Service
- Implementing Dynamic Filtering for RESTful Service
- Versioning RESTful Services - Basic Approach with URIs
- Versioning RESTful Services - Header and Content Negotiation Approach
- Implementing Basic Authentication with Spring Security

Module 6: JPA in Spring Boot

- Introduction to JPA
- Object Relational Impedence Mismatch
- World before JPA - JDBC, Spring JDBC and myBatis
- Introduction to JPA
- Creating a JPA Project using Spring Initializer
- H2 Database URL
- Defining a JPA Entity - User
- Defining a Service to manage the Entity - UserService and EntityManager
- Using a Command Line Runner to save the User to database.
- Magic of Spring Boot and In Memory Database H2
- Introduction to Spring Data JPA
- More JPA Repositories

Module 7: Connecting Restful Services with JPA

- Overview of Connecting RESTful Service to JPA
- H2 Database URL
- Creating User Entity and some test data
- Updating GET methods on User Resource to use JPA
- Updating POST and DELETE methods on User Resource to use JPA
- Creating Post Entity and Many to One Relationship with User Entity
- Implementing a GET service to retrieve all Posts of a User
- Implementing a POST service to create a Post for a User
- Richardson Maturity Model
- RESTful Web Services - Best Practices

Module 8: Spring Booth Security

- Basics
- Basic Authentication
- Form Based Authentication
- Authorization
- Role Based Access Control
- Attribute Based Access Control
- LDAP Based
- SSL Security
- TLS Security

Module 9: Database Concepts

- Spring JDBC
- Database to CSV
- Spring Batch
- Flyway Database Migration
- Liquid Database Migration
- Flyway vs Liquid

- Hikari Connection Pool

Module 10: Core Services

- Spring Boot AOP
- Spring Boot Cache
- Guava Cache integration
- Caffeine Cache
- EH Cache
- MultiResourceItemReader
- Spring MVC vs JAX-RS
- Spring Boot with Jersey
- Junit Integration
- Rest Integration Test Cases

Module 11: Micro Services

- Micro Services Introduction
 - Principle and Characteristics
 - Use cases and Benefits
 - Challenges
 - Design standards
 - Pitfalls
- Micro Services Communication
 - Synchronous
 - Asynchronous
- Micro Services Design Considerations
 - Micro Services per JVM?
 - Micro Services share the data stores?
 - Micro Services Transaction boundaries
 - User Interfaces integration with Micro Services
 - Challenges in Micro Services implementation

Module 12: Spring Cloud

- Spring Cloud
 - Introduction
 - Cloud Architecture
 - Cloud application benefits
- Spring Cloud Config
 - Introduction
 - Setup version control repository
 - Integration with repository
- Netflix
 - Introduction
 - Eureka Server & Eureka Client
 - Feign Client
 - Ribbon

- Fault Tolerance Concepts
 - Circuit Breaker Pattern
 - Hystrics Concepts, Hystrics Dashboard
- API Gateway
 - Introduction to ZUUL
 - Design standards
 - Integration
- Messaging Queue Concepts (Cloud Bus)
 - Apache KAFKA
 - RabbitMQ
 - JMS
- Oatuh2 Concepts
 - Client Types
 - Protocol End Points
 - Grant Types
 - Implantation with Token Based
 - JWT Tokens
- Swagger API
 - Introduction
 - Integration
- Cloud Hosting
 - Pivotal Cloud Foundry Account Setup
 - Hosting to Pivotal
 - AWS Account Setup
 - Hosting to AWS
 - Enabling cloud features like load balancing, security

Placement Guide

- What is an Interview?
- Tips to clear an Interview
- Common Interview questions and answers
- Spring Boot Interview Questions and Answers
- Resume Building Guide
- Attempt for Spring Boot Global Certification Exam
- Start applying for Jobs