Full Stack Java Developer

The Full Stack Java Developer Job Guarantee Program offers a comprehensive set of software development skills. This one-of-a-kind industry curriculum will help you learn the entire Full Stack Java Development process. Create industry-ready projects and be prepared to land opportunities in top organisations.

Duration : Language : English Price : 17700

What you will learn?

- Fundamentals of Programming
- Core Java (Detailed)
- JDBC
- JEE (Servlets, JSP, and Thymleaf)
- Hibernate and JPA specifications
- Spring Core
- Spring Boot
- Spring JDBC
- Spring ORM
- Spring Data JPA
- Spring AOP
- Spring MVC
- Spring REST
- Microservices and Realtime tools(Maven, Gradle, Log4J, Junit, Splunk, Putty, Jacacco)
- Docker and Kubernetes
- Agile and Scrum
- Git and Github
- HTML and CSS
- Javascript
- React js
- SQL - Mysql
- NoSQL - MongoDB

Features

- Full stack Java Developer certification
- Job guarantee Program

- Online Instructor-led learning: Live teaching by instructors
- 250+ hours live interactive classes.
- Every week doubt clearing session after the live classes.
- Lifetime Dashboard access
- Doubt clearing through mail and discussion forum
- Quiz in every module
- A live project with real-time implementation
- Resume building
- Career guidance
- Interview Preparation
- Regular assessment
- Mock Interview
- Course resources
- On demand recorded videos
- Practical exercises
- Quizzes
- Assignments
- Course completion certification

Requirements

- System with Internet Connection
- Interest to learn
- Dedication

Course Curriculum

Induction of Course

- Introduction to course and Q&A

Git and Github

- Git foundation
- Setting, maintaining and tracking git repos
- Git snapshots
- Git for team management
- Git branches
- Git merging

- Git and Github ecosystem

Software Installation

- Download and Install Java
- Download and Install Eclipse
- Download and Install Visual Studio Code

Fundamentals of Java

- Introduction to Programming
- Basic Understanding of a Computer
- Basic feature of Java
- Main method
- Classes and Objects(Basics)
- Statically typed vs Dynamically typed Programming Language
- Variables and Data type in Java
- Naming Convention
- Identifiers

Operators and Loops

- Operators in Java
- Incrementation and Decrementation
- Condtional statement
- Ternary operator
- Switch case
- Loops intro
- for - while - do while
- More on loops
- Scanner class and User input in Java
- Pattern programs
- Nested loops

Oops Fundamentals

- Object creation
- Instance variable vs Local variables
- Methods with memory maps (JVM data areas)
- Method overloading

Mini Project

- Guesser Game Project

Array in Java

- Why array?
- What is an Array?
- How to create an array
- 1D, 2D, 3D and Regular Array & Jagged Array with memory map
- Buffer overrun and ArrayIndexOutOfBoundsException
- Disadvantages of Array in Java
- Few basic programming questions
- Bubble Sort
- Selection Sort
- Merge Sort
- Linear Search
- Binary Search

String in Java

- String Introduction
- Types of string
- Immutable string
- Ways to compare and memory map String constant pool
- Inbuilt methods in String class
- Concatination
- Few Programming questions discussion
- (Reversing String, Palindrome, Anagram, Pangram ....)
- Mutable String
- String Buffer vs String Builder
- Inbuilt Methods

Static Keyword

- Static keyword
- Class loading
- Execution of a Java Program
- static variables, static methods, static block
- Differences b/w Non static and static

Encapsulation

- Need of Encapsulation
- What is Encapsulation?
- Private members
- Shadowing problem and this keyword
- Setters & Getters
- Constructor
- this()

Inheritance

- Inheritance introduction
- extends keyword
- Types of Inheritance
- Important key points(5 keypoints)
- Inherited methods, Overidden methods, Specialized methods
- Rules to override method

Polymorphism and Abstraction

- What is polymorphism ?
- How to achieve polymorphism
- Runtime vs Compile time polymorphism
- Abstract keyword and Abstraction
- Abstract class and Abstract method

Final keyword in Java

- final class
- final variable
- final method

Interface

- What is interface
- Need of Interface
- Different use cases of Interface
- Abstract vs interface
- Additional features of Interface

Lambda Expression

- Functional Interface
- What is Lambda Expression
- Different ways to create Lambda Expression
- Lambda Expression excercises

Exception Handling

- What is an Exception?
- How to handle Exception (try catch)
- Multiple catch block
- Handling vs Ducking an Exception
- Hierarchy of an Exception class
- throw & throws keyword and Custom Exception
- try with Resources

Core Java Project

- Assignment with mentor guidance - Food Delivery App

Multi-threading

- What is Thread & Need of multiple Threads
- How to create multiple Threads
- run() method
- Race condition
- Different states of Thread
- Dead lock

Collection in Java

- Why Collection ?
- ArrayList
- LinkedList
- PriorityQueue
- ArrayDeque
- TreeSet
- HashSet
- LinkedHashSet
- Collection Hierarchy
- Map

- Map heirarchy
- Stream API in Java

Annotations in Java

- Enums
- What is Annotation
- In Built Annotation
- Custom Annotation

File Handling in Java

- Input Stream
- Output Stream
- File Operation in Java
- Serialization
- Deserialization

SQL - MySQL

- Basic Concepts of Advantages of DBMS
- Exploring Relational DBMS
- E-R Modeling and Diagram
- Normalization
- Introduction to SQL
- DDL and DML Statements
- Working with Queries (DQL)
- CRUD operations
- Aggregate Functions
- Joins and Set Operations
- Working with Constraints

MongoDB

- What is mongoDB
- How does mongoDB works
- What is mocha and need of mocha in mongodb
- Big umbrella of MongoDB
- How to install mongoDB on MAC
- How to install mongoDB on Windows

- Create and Read operation in MongoDB
- ObjectID and BSON in mongoDB
- CRUD operations in mongoDB
- UpdateOne and DeleteOne in #mongoDB
- UpdateMany and deleteMany in mongoDB
- Database issues with Update in mongodb
- Getting more data in #mongodb
- Understanding objects structure in mongoDB
- What is schema in mongoDB

JDBC

- Steps followed to write JDBC Code
- Usage of Statement Object
- Usage of Prepared Statement
- Types of Driver available
- Application using Statement and PreparedStatetement

Project - JDBC

- CRUD operation applicationin layered approach of Student table using Factory Desgin Pattern

HTML and CSS

- Collecting and installing developers tool
- Structuring the files and creating first file
- Text tags
- List items
- Divisions and Spans
- Images and links
- Challenge for links on images and solution
- Tables in HTML
- More about forms in HTML
- Comparing HTML 4 semantics with HTML 5
- Introduction to css and where to write it
- Solving the color selection problem
- Comming soon template and backgrounds
- Box model and centering text
- Google fonts and font awesome

- Styling the links
- Classes and ID in CSS
- Designing a navigation bar from scratch
- Color palletes and canva for design
- Gradients in css
- Check through css
- box sizing in css

JavaScript

- What are JavaScript engines
- What ES version of JavaScript is good for us
- Variable and datatypes in JavaScript
- Operators in JavaScript
- What are conditionals in JavaScript
- Logical conditional Login in JavaScript
- Ternary operator in JavaScript
- Switch for role-based access in JavaScript
- Basics of functions in JavaScript
- Functions in variable User Role in JavaScript
- Understand the context in JavaScript
- Code hoisting in JavaScript
- Scope chaining in JavaScript
- Light intro to THIS in JavaScript
- Maps in JavaScript
- Classes and module exports in JavaScript
- Private props getters and setters in JavaScript
- Inheritance in JavaScript
- Event loop Will JavaScript wait
- Promise async and await in JavaScript

React js

- What is react and myths
- Tools that we need
- Introduction of Virtual DOM.
- Difference between JS and JSX.
- React Components overview

- Containers and components
- Child Components
- Namespaced components
- JavaScript expressions available in JSX
- Node setup
- How to use NPM?
- How to create package.json and purpose of it
- Best IDE for React JS and How to write optimized code in React JS?
- React JS browser plugins overview.
- Create a React component with JSX template.
- How to create Nested Components?
- What is React JS render?
- React Props overview.
- Introduction of Props validation with data types.
- Flow of States, Initialize states and update states.
- Lists of Form components.
- Setup Controlled and Uncontrolled form components.
- Control Input elements.
- How to set default values on all formats of Input elements.

React JS Form validations.

- How to write Styles?
- Initial Render
- Props Change
- Stage Change
- Component willMount
- Component didMount
- Component Unmount
- Overview of a single-page application.
- How is React Router configured?
- Background of Router
- How Should Conditional Statements Be Handled in JSX?
- onBlur, onKeyUp, onChange and other useful primary events in React JS.
- How to Sharing events between the components?
- Introduction to styled components
- Styling the application using styled component

- How to Load the router library?
- Configure the React Router?
- How to Pass and receive parameters?
- Understanding Hooks
- The useState hook
- Side effects using the useEffect hook
- The useContext hook
- The useReducer hook
- Writing your own hook
- The React ecosystem

Servlet

- Types of application
- Client Server Architecture
- Different types of Server a. web server b. application server
- Need of Servlet and Different ways of Creating a Servlet
- Configuring Servlet in
- XML and Annotation support
- Difference b/w ServletConfig vs ServletContext object
- HttpServletRequest,HttpServletResponse,RequestDispatching
- SessionTracking Mechanism
- HttpSessionTracking
- Cookie
- URL ReWriting
- Hidden form Field
- Filters,Listeners and One CRUD app using MVC Design pattern
- Need of JSP, Usage of JSP, Implicit Objects
- Type of Directives
- Expression Language, JSTL Tags
- MVC CRUD APP using Servlet, JSP

Project - JEE

- Building CustomerRelationship manager System using JDBC,Servlets and JSP and JSTL

Introduction to ORM(Hibernate and JPA Specifications)

- Drawbacks of JDBC

- Hiberante
- Advantages of Hibernate compared to JDBC
- Introduction.
- ORM (Object Relational Mapping)
- Configuration xml file and Mapping xml file along with dtds.
- Hibernate architecture
- Installation and Directory Structure
- Hibernate Data Types.
- First Application using Hibernate.
- Hibernate API
- CRUD operations
- Primary key Generators
- Hibernate Query Language (HQL)
- Native SQL
- Criteria API
- Inheritance in Hibernate
- Relations
- (one to one, one to many, many to one, many to many)
- Caching
- Connecting with Multiple Databases
- Integrating Hibernate with Servlets,JSP and with Spring
- Hibernate Annotations
- Performing BLOB/CLOB operation, Insertion of Date and Time to Database
- Performing Object versioning TimeStamping and life cylce events of hibernate
- ConnectionPooling in hibernate

Project - Hibernate

- Building CustomerRelationship manager System using ORM,Servlets and JSP and JSTL

SPRING BOOT

- What is Spring Framework
- What is Spring Boot
- Differences between Spring & Spring Boot
- IOC container
- Dependency Injection a) Setter Injection b) Constructor Injection c) Field Injection
- Stereotype Annotations a) @Component b) @Service c) @Repository d) @Controller e) @Indexed

- Spring Boot Overview
- Pros & Cons of Spring Boot
- Approaches to create Spring Boot Application
- Spring Initializer (start.spring.io)
- Spring Starter Wizard in STS IDE
- Introduction to Spring Boot Starters
- Spring Boot Parent Starter
- Spring-boot-starter
- Spring-boot-starter-web
- Spring-boot-starter-webflux
- Spring-boot-starter-data-jpa
- Spring-boot-devtools
- Spring-boot-starter-mail
- Spring-boot-actuator
- Spring-boot-starter-test etc.
- What is Start Class in Spring Boot
- @SpringBootApplication annotation internals
- SpringApplication.run(..) method internals
- Spring Boot Application Boot strapping
- AutoConfiguration in Spring Boot

SPRING DATA JPA

- What is Persistence Layer
- Best practises to follow in persistence layer
- ORM Basics
- Spring Data JPA Introduction
- Differences between Spring ORM and Spring Data
- CurdRepository introduction
- CurdRepository methods for DB operations
- Custom findByXXX method syntax
- Custom Queries Execution in Data JPA
- JpaRepository introduction
- JpaRepository methods for DB operations
- Pagination Using Data JPA methods
- Sorting Using Data JPA Methods

- Query By Example Executor
- Generators
- Custom Generators in Spring Data
- Embedded Database Introduction
- Embedded Database vs External Database
- Application Development using Embedded Database (H2)
- Application Development Using MYSQL Database
- Application Development Using PostGreSQL Database
- Application Development Using MongoDB
- profiles in springboot

SPRING WEB MVC

- Spring Web MVC Introduction
- Spring Web MVC Advantages
- Spring MVC Architecture
- Introduction to Front Controller
- Controllers
- Handler Mappers
- View Resolvers
- Web Application development using Spring Boot
- Embedded HTTP Servers Introduction
- a) Embedded Tomcat Server b) Embedded Jetty Server c) Embedded Undertow Server
- Making Jetty as Default server
- Web Application Deployment in External Server
- Sending Data From UI to Controller
- a) Query Param b) Path Param
- Sending Data From Controller to UI a) Model b) ModelAndView
- @RequestBody annotation 38) @ResponseBody annotation
- Introduction to Spring MVC Form Tag library
- Form Based application development using Spring Boot
- Thymeleaf Introduction
- Web Application with Thymeleaf
- Sending Email using Spring Boot
- Exception Handling in Spring Boot Web Application
- Spring Boot Actuators

- a) Health b) Info c) Heapdump d) Theaddump
- e) Beans f) Httptrace g) Mappings h) Shutdown etc
- Unit Testing for Spring Boot Applicationusing Junit with Mocking
- Code Coverage using Jacocco

SPRING REST

- Distributed Applications
- Distributed Technologies
- SOAP vs REST
- RESTful Services Introduction
- REST principles
- XML
- One Time operations
- Run Time Operations a) Marshalling b) Un Marshalling
- JAX-B Introduction JAX-B Architecture
- Applications development with JAX-B
- JSON Introduction
- XML vs JSON
- JACKSON API
- Converting Java object to JSON and vice versa using Jackson API
- GSON API
- Converting Java Object to JSON and Vice Versa using GSON API
- HTTP Protocol Details
- HTTP Methods a) GET b) POST c) PUT d) DELETE
- HTTP Status Codes
- @RestController
- @RequestBody
- @ResponseBody
- @RequestParam
- @PathVariable
- MediaTypes
- Consumes
- Produces
- Accept Header
- Content-Type head

- REST API Development using Spring Boot
- POSTMAN
- SWAGGER & SWAGGER UI
- Exception Handling in REST API
- REST Security
- a) HTTP Basic Auth
- b) JWT
- c) OAuth2.0
- Mono Objects
- Flux Objects
- REST Client Introduction
- RestTemplate
- WebClient
- RestTemplate vs WebClient
- Reactive Programming
- Synchronous vs Asynchronous Calls
- Apache Kafka Integration with Spring Boot
- Redis Cache Integration with Spring Boot

Spring Boot Projects

- Building Student management System using SpringBoot
- Building CustomerRelationship manager System using SpringMVC and Thymleaf
- Working with TicketManagement application using Spring datajpa and Spring ReSt with swagger integration

Docker

- Docker & its architecture
- Docker as a service
- Docker CLI
- Docker Volumes
- Dockerizing a web application

MICROSERVICES

- Monolith Architecture case study
- Monolith Application Deployment Process
- Load balancer (Cluster) case study

- Load Balancing Algorithms
- a) Round Robin
- b) IP Hashing
- c) Sticky Session
- Monolith Architecture Drawbacks
- Micro services Introduction
- Micro Services Advantages
- Micro Services Dis-Advantages
- Micro Services case study
- Identifying Micro services boundaries
- Micro services Architecture
- Micro services Development

Agile and Scrum

- What is Agile?
- What is Scrum?
- Benefits of Agile
- Scrum Artifacts

Final Project 1

- Building StockMarket API integration with Eureka Client and hosting in PCF

Final Project 2

- Buidling BookStock AP integration with MongoDB and making it as Eureka Client with swagger integration

Final Project 3

- Capstone project of Insurance application which holds microservices and React integration