GIRI'S TECH HUB Pvt Ltd

...Talent Meets Opportunity

5400+ Placements (SINCE 2018)

14 LPA Highest Package

2000+ Campus

1000+ Clients

Professional Certificate Program

Diploma in Java Full Stack Development with Generative AI & Agentic AI

Pune's Trusted IT Training & Placement Institute

become a

Java Full Stack Developer

In Just 9 Months

> Job-Ready with The Giri's Tech Hub
> Java Full Stack with GenAI Certification program

Mr. Adinath Giri

Founder & Java Expert

GIRI'S TECH HUB Pvt Ltd
...Talent Meets Opportunity

CONTENTS

Mapping your Coding Future!

01 Success Stories!!

02 Java Full Stack Overview

03 Course Journey

04 Course Syllabus

05 Tools to Master

06 Project Training

07 Career Readiness Program

08 Placement Preparation

09 Program at Glance!!

10 Meet your Mentors!!

11 Our Alumni Work

12 Go for it!!

Success Stories!!

Meet Our Alumni

Thousands of our alumni are placed in top IT companies, reflecting their excellence and professional expertise.

SHREYASH KAWANE

Software Engineer

14 Lpa

Branch : BE (ENTC Engineering)
College : Pune Institute of Computer Technology

DIVYA CHAVHAN

Software Developer

Cyncly 10 Lpa

Branch : B.Sc (CS)
College : Rani Laxmibai Mahavidyalaya, Parola

RAVINDRA DANE

Software Engineer

CISIVE 9.0 Lpa

Branch : BE (Computer Engineering)
College : Sandip Institute of Engineering and Management, Nashik (SIEM)

PRIYA CHOUGALE

Software Engineer

VALUEMOMENTUM 9.0 Lpa

Branch : BE (Computer Engineering)
College : Vishwakarma Institute Of Information Technology

PRIYANKA PATIL

Revit Developer

CUNDALL 8.0 Lpa

Branch : B.Tech (CE)
College : Gangamai College of Engineering, Dhule

SARFARAJ BOBADE

Software Engineer

zinnia 7.5 Lpa

Branch : BE (Computer Engineering)
College : D Y Patil College Of Engineering, Ambi Talegaon Tal Maval.

TUSHAR SHINDE

Software Engineer

FINASTRA

7.3 Lpa

Branch : BE (Mechanical Engineering)
College : Vishveshwarya Technical Campus, Miraj.

RUTUJA AMRUTKAR

Software Engineer

7.0 Lpa

Branch : BE (Computer Engineering)
College : Bharati Vidyapeeth College of Engineering, Lavale, Pune.

LOKESH WAGHMARE

Software Engineer

AXIS BANK

7.5 Lpa

Branch : B.E (Civil Engineering)
College : Anjuman College of Engineering and Technology, Nagpur.

SAHIL MUJAWAR

Software Engineer

RAJA SOFTWARE LABS peace of mind

7.0 Lpa

Branch : BE (Civil Engineering)
College : Sharad Institute of Technology College of Engineering, Yadrav.

ARJUN WADURKAR

Software Engineer

RAJA SOFTWARE LABS peace of mind

7.0 Lpa

Branch : BE (Computer Engineering)
College :Dr. D Y Patil College of Engineering & Innovation, Varale, Talegaon, Pune

VEDANT NAVARKAR

Software Engineer

RAJA SOFTWARE LABS peace of mind

7.0 Lpa

Branch : BE (Computer Engineering)
College : Modern Education Society's Wadia College of Engineering, Pune

ABHIJIT DESHMUKH

Software Engineer

Loylty Rewardz

7.0 Lpa

Branch : BE (Civil Engineering)
College : Pankaj Laddhad Institute Of Technology And Management Studies, Buldhana.

ONKAR CHAWARE

Software Engineer

LogicMonitor

7.0 Lpa

Branch : BE (Computer Engineering)
College : Bapurao Deshmukh College of Engineering, Sewagram, Wardha.

ADITYA WADIKAR

Software Engineer

RAJA SOFTWARE LABS peace of mind

7.0 Lpa

Branch : BE (Mechanical Engineering)
College : Sinhgad college of Engineering, Vadgaon, Pune.

Java Full Stack Overview

Gain in-demand Java Full Stack skills from basics to advanced through hands-on projects.

Frontend Technologies

1. HTML
2. CSS
3. JavaScript
4. BootStraps
5. Modern Frameworks
    1. ReactJs
    2. Redux
    3. NextJs

Backend technologies

1. Core Java +DSA
2. Adv. Java JDBC
3. Servlet
4. JSP
5. Spring
6. Spring Boot
7. Hibernate
8. RESTful APIs
9. Microservices

Databases

1. SQL & PLSQL

Dev Tools

1. GIT
2. JIRA
3. JUNIT
4. MAVEN
5. Docker
6. Jenkins
7. Mocketo
8. Bitbucket
9. Kubernetes
10. SonarQube
11. Logging Tools
12. Travis CI CI/CD

Course Journey

Learning Path of Java Full Stack

Step-by-step journey from fundamentals to advanced java full stack skills

01 Core Java + DSA

02 Jdbc, Servlet, Jsp

03 Spring & Spring Boot, Hibernate, Micro Services

04 SQL/PLSQL

05 HTML5, CSS3, JavaScript, Bootstrap, React.js

06 Live Project Training

Inside the Program

Uncover the Program Modalities

- Offline & Zoom Sessions
- Hands-on Practical Sessions
- Soft Skills Sessions
- Aptitude & Logical Reasoning Sessions
- Mock Interviews with Industry Experts
- Console-Based Projects
- Real-Time Live Projects

Course Syllabus

In-Depth Syllabus – Java Full Stack

Front End Development Module

Introduction of Java

- Introduction of JAVA
- Installation JDK
- Creating Simple application
- Data Types
- Variables
- Constant
- Operator & Its Types
- Operator Priority

- 25 Trainer-solved problems,
- 50 Lab assignments,
- 30 MCQs

Decision making Statement

- Introduction of Decision Making
- Simple If Statement
- If else statement
- Else if ladder
- Nested if statement
- Switch Statement

- 25 Trainer-solved problems,
- 50 Lab assignments,
- 10 CodeChef problems
- 30 MCQs

Control Statement

- Introduction about control statement
- For loop
- Nested for loop
- While loop
- Do while loop
- Break & continue keyword

- 15 (Series & Numbers) trainer-solved
- 15 start patterns trainer-solved
- 10 number patterns trainer-solved
- 10 alphabetical patterns trainer-solved
- 30 (Number and series) lab assignments,
- 50 (Pattern problem) lab assignments

Function & Function Recursion

- What is Function & Why use it
- Types of Function
- How to return value from function
- How to pass parameter to function
- What is function recursion
- How to manage function recursion
- Local Variable concept
- Limitation of function recursion
- Dynamic programming introduction
- Types of dynamic programming

Time Complexity & Space complexity

- What is Time complexity
- Important of Time complexity
- How to calculate time complexity
- Types of Complexity like as O(N), O (log n), O (n log n),O(n^2) etc.
- Time complexity comparison chart
- What is space complexity
- Types of space complexity
- Common Space Complexities like as O(1), O(n), O(log n), O(n^2)etc.

- 50 MCQ for finding time complexity and space complexity

Array with simple operation

- What is array & why use it
- Single and multi-dimension array
- Jagged Array concept
- Linear & Binary Search
- Stack
- Queue & DQueue

- 20 Trainer-solved problems
- 30 Lab assignments
- 30 MCQs

Array with problem solving approach

- Brute Force Approach
- Two pointer approach
- Sliding Window Approach
- Dynamic Programming
- Sliding window + DQueue
- Frequency Counting / Hashing
- Rearrangement & Transformation
- Bit Manipulation Based
- Merging Arrays
- Subarray and Subsequence Problems

- 20 Assignments solved by trainer
- 30 Lab assignments

Array with Sorting Technique + complexity comparison

- What is sorting
- Selection sort
- Bubble sort
- Insertion sort
- Merge sort
- Quick Sort
- Tim Sort
- Heap Sort

- 15 Trainer solved problems
- 30 Lab Assignments

Classes and objects

- What is class & why use it
- What is object and reference
- Static, instance and local variable
- Variable argument
- This reference
- Array as parameter to function
- Array of objects concept
- Nested classes

- 15 Assignments solved by trainer
- 30 Lab assignments

Constructor

- What Constructor & Why use it
- Implicit constructor concept
- Types of Constructors like as default, overloaded, parameterized etc
- Constructor chaining
- Difference between function and constructor
- Singleton & utility class

- 15 Trainer solved problems
- 30 Lab Assignments

Inheritance

- What Inheritance & Why use it
- Types of Inheritance
- Super constructor
- Super keyword
- Final keyword.
- Method overriding
- Abstract keyword
- Dynamic polymorphism
- Loose coupling & Tight coupling
- Interface
- Method hiding
- Method with covariant return type
- JDK 1.8 feature with interface
- UP Casting & Down Casting

- 15 Trainer-solved problems
- 30 Lab assignments

Exception Handling

- What is exception & why use it
- Types of Exception
- Exception Hierarchy
- Exception handling keyword like as try, catch, finally, throw and throws
- Handling checked exception
- User defined exceptions
- Handling user defined exception
- JDK 1.7 exception Feature
- Difference between throw and throws
- Difference between catch and exception

- 10 Trainer-solved problems
- 15 Lab assignments

Wrapper classes

- What is Wrapper class & need
- Types of Conversion
- Auto boxing
- Auto unboxing
- Number class & Its method parseXXX(), xxxValue(), valueOf() etc
- String
- StringBuffer, StringBuilder

- 15 Assignments solved by trainer
- 30 Lab assignments

Advance JavaScript (50%) content

- Arrow function
- Rest operator & Spread operator
- Validation using JavaScript
- Promise, Async and Await
- Event loop concept
- JSON & AJAX Concept
- Fetch API
- Function currying concept

- 20 Trainer solved problems
- 20 Lab Assignments

Non-Linear DS

- Tree – Binary Search, Complete Binary Tree, Max Heap, Min Heap
- Tree Traveling – In order, preorder and post order
- Graph - Directed Graph and Undirected Graph
- BFS, DFS, Dijkstra's Algorithm
- Hash Tables / Hash Maps / Hash Sets
- Unordered Set / Map

- 15 Assignments solved by trainer
- 30 Lab assignments

File Handling

- What is IOStream & Why use it
- Write, Reader, Stream classes
- Writing Data in file using Writer classes
- Writing Data in file using Stream classes
- Reading data from file using Reader classes
- Reading data from file using Stream classes
- Serialization concept
- Deserialization concept

- 15 Trainer solved problems
- 30 Lab Assignments

Multithreading

- What is Thread & its benefit
- Ways to implement thread
- Thread implementation using thread class
- Synchronization and asynchronization concept
- wait(), notify() and notifyAll() methods
- Conditional and unconditional methods
- Thread methods like as join(), start(), isAlive() etc
- Thread pooling concept
- Executor services
- Thread implementation using runnable

- 10 Trainer-solved problems
- 15 Lab assignments

Collection Framework & Map Framework + Object

- Introduction Of Collection Framework
- Collection Hierarchy
- Collection, List, Set, Queue
- Vector, ArrayList, Stack, LinkedList, HashSet, LinkedHashSet, TreeSet, Dqueue, PriorityQueue
- Collections class & its method
- Comparable, Comparator interface
- Map interface & it's working
- HashMap, LinkedHashMap, TreeMap
- Object class & its method
- Difference between catch and exception

- 20 Trainer-solved problems
- 30 Lab assignments

JDK 1.8

- Interface changes
- Functional interface
- Lambda interface
- Consumer, Predicate, Supplier, Function Interface etc
- Stream API
- Optional class, Date and Time API
- Method references

JDK 1.9

- Module concept
- JShell & private methods in interface
- Try with resource enhancement
- Factory methods List.of(), Set.of(), Map.of() etc
- Stream API enhancement like as takeWhile(), dropWhile(), iterate()
- Optional API enhancement like as ifPresentOrElse(), or(), stream()

JDK 11

- var variable in lambda
- New string methods
- New Files API methods
- Optional API enhancement

JDK 17

- Sealed classes
- Pattern matching switch
- RandomGenerator, HexFormat
- Foreign Function & Memory API(Incubator)

JDK 24 & 25

- Sealed classes
- Pattern matching switch
- RandomGenerator, HexFormat
- Foreign Function & Memory API(Incubator)

Database concept with SQL using MYSQL

- What is DBMS & SQL & its benefit
- SQL all basic command like as DDL, DML, TCL, DCL etc
- Where, group by, having, like, IN, between, any, all, exits etc
- Constraints – not null, unique, primary, foreign, default, auto_increment, check, on delete cascade, on update cascade, on delete set null etc
- Join and its type like as inner join, left join, right join, outer join etc
- Normalization – FIRST NF, Second NF, Third NF, 3.5 NF, Fourth form etc
- Subquery, view, index, procedure
- Trigger, function, ACID property

Spring Framework

1. Spring Core

- Introduction to Spring core
- Types of dependency like as setter, constructor, object dependency, map, list dependency etc
- Spring application using XML and annotation
- Cover some annotations like as @Configuration, @Value, @Bean, @ComponentScan, @Qualifier, @Primary, @Scope, @PostConstructor, @Service, @Repository, @Controller, Component etc.
- object dependency and auto wiring
- bean scope and bean life cycle
- spring container and its type
- spring container and its differences

- 15 Trainer-solved problems
- 20 Lab assignments

2. Spring JDBC

- Introduction about spring jdbc
- Connection establishment using DataSource using XML and annotation based
- CRUD APP using JdbcTemplate like as insert, delete, update, select
- Batch Update concept
- RowMapper, PreparedStatementSetter etc
- Calling Stored procedure using SimpleJdbcCall registering IN, OUT and INOUT parameter
- Transaction management using @Transactional annotation
- Exception Handling like as DataAccessException etc
- Difference between catch and exception

- Mini Project using Spring core + spring JDBC in class room
- Mini Project as lab assignment

3. Spring MVC

1. Introduction to Spring MVC

- What is Spring MVC?
- Spring MVC Architecture
- Role of DispatcherServlet
- Flow of a typical Spring MVC request
- Comparison with other MVC frameworks

2. Spring MVC Setup

- Project structure (with Maven or Gradle)
- Required dependencies (Spring Web, Spring Context)
- Java-based configuration (no XML)
- Configuring DispatcherServlet via WebApplicationInitializer
- Defining @Configuration classes

3. DispatcherServlet & Request Flow

- Role and life cycle of DispatcherServlet
- HandlerMapping and ViewResolver
- Component scanning with @ComponentScan
- Web application bootstrap using @WebMvcConfigurer

4. Spring MVC Setup

- Creating controllers with @Controller
- @RequestMapping and its variants (@GetMapping, @PostMapping, etc.)
- Path variables (@PathVariable)
- Request parameters (@RequestParam)
- Handling query strings and matrix variables
- Returning view names and response bodies

5. Model and View

- Using Model, ModelMap, and ModelAndView
- Sharing data between controller and view
- Redirect and forward in Spring MVC
- Using Flash attributes with RedirectAttributes
- Passing model data to views

6. View Technologies

- Overview of supported view technologies
- JSP + JSTL
- Thymeleaf
- FreeMarker (optional)
- Setting up view resolvers
- View templates and rendering

7. Form Handling in Spring MVC

- Binding form data to Java objects using @ModelAttribute
- Handling GET and POST forms
- Prepopulating form fields
- Working with dropdowns, checkboxes, and radio buttons
- Form re-submission and Post/Redirect/Get (PRG) pattern

8. Data Validation & Binding

- Bean Validation with JSR-380 (Hibernate Validator)
- Using @Valid and BindingResult
- Custom validators
- Field-level error messages
- Global error handling
- Formatting numbers and dates

9. Exception Handling

- Using @ExceptionHandler in controller
- Global exception handling with @ControllerAdvice
- Handling HTTP errors (404, 500)
- Returning custom error pages or JSON error responses

10. RESTful Web Services with Spring MVC

- Creating REST controllers using @RestController
- Returning JSON/XML responses with @ResponseBody
- Consuming path variables and request parameters
- Handling HTTP methods: GET, POST, PUT, DELETE
- Content negotiation and producing/consuming media types
- Exception handling in REST (@RestControllerAdvice)

11. Spring MVC with Database (JDBC/JPA)

- Integrating with Spring JDBC or Spring Data JPA
- Injecting DAO or repository into controllers
- Using @Service and @Repository layers
- Displaying data from DB in views
- Form submissions that persist data

12. Session & Cookie Management

- Working with @SessionAttributes and SessionStatus
- Accessing session and cookies via HttpSession, @CookieValue
- Custom session timeout handling

13. Security in Spring MVC (Optional)

- Introduction to Spring Security
- Protecting endpoints and views
- Form-based authentication
- CSRF, CORS, and session management

- 15 Trainer-solved problems
- 15 Lab assignments
- Note: 64 hours project hackathon in project lab 8 hours per day workload with 8 days

Spring Boot with Micro services & AI

1. Spring Boot

1. Spring Boot Fundamentals

- Spring Boot architecture
- Auto-configuration & Spring Boot Starters
- Spring Initializer
- application.properties vs application.yml
- Profiles (@Profile)
- Logging & external config

2. Spring Boot Web

- REST APIs with @RestController, @RequestMapping
- JSON serialization with Jackso
- Exception handling (@ControllerAdvice, @ExceptionHandler)
- HTTP status codes and custom error responses

3. Spring Boot with Databases

- JDBC and JPA overview
- Spring Data JPA
- Repositories, derived queries
- Paging and sorting
- Entity relationships (OneToMany, ManyToOne, etc.)
- H2, MySQL/PostgreSQL integration

4. Validation and Data Binding

- @Valid, @NotNull, @Size, etc
- BindingResult and error responses
- Custom validators

5. Testing in Spring Boot

- Unit testing with JUnit 5 + Mockito
- Integration testing with @SpringBootTest
- Controller testing with @WebMvcTest
- Using Testcontainers (optional)

2. Micro Services

1. Microservices Architecture Overview

- Monolith vs Microservices
- Key concepts: loose coupling, bounded context, scalability
- 12-Factor App principles
- Microservices communication patterns (sync/async)

2. Creating Microservices with Spring Boot

- Splitting business domains into services
- Example domains: Product, Order, Payment, Inventory
- Best practices for service packaging and structure

3. Inter-Service Communication

- REST-based communication using RestTemplate (legacy)
- Feign Client (@FeignClient) – preferred
- Load balancing with Spring Cloud LoadBalance

4. Service Discovery with Eureka

- Eureka Server setup
- Registering microservices as Eureka clients
- Using Eureka with Feign

5. API Gateway with Spring Cloud Gateway

- Setting up Gateway
- Routing, path rewriting
- Load balancing via discovery
- Pre/post filters (logging, auth, rate limiting)

6. Centralized Configuration with Spring Cloud Config

- Spring Cloud Config Server setup
- Storing configs in Git
- Dynamic configuration refresh with Actuator + Bus

7. Fault Tolerance and Resilience

- Circuit breaker with Resilience4j
- Retry, TimeLimiter, RateLimiter
- Fallback methods
- Bulkhead and Isolation

8. Distributed Tracing and Logging

- Centralized logging with ELK or EFK stack (ElasticSearch + Logstash + Kibana / Fluentd)
- Tracing with Spring Cloud Sleuth + Zipkin
- MDC and correlation IDs for logs

9. Security in Microservices

- Spring Security basics (@EnableWebSecurity, filters)
- JWT Authentication & Authorization
- Role-based access control
- OAuth2 (optional, for advanced cases)
- API Gateway as authentication entry point

10. Monitoring and Observability

- Spring Boot Actuator endpoints
- Custom health checks
- Metrics with Micrometer + Prometheus
- Dashboards with Grafana

11. Service Versioning & Communication Contracts

- API versioning (URI vs headers)
- Using Swagger/OpenAPI
- Contract testing with Spring Cloud Contract (optional)

3. Generative AI

- Introduction to Spring AI
- Core Features of Spring AI
- Integrating OpenAI with Spring Boot
- Setting up Spring AI with Free LLM (Ollama)
- Chat Client & Chat Model APIs
- Working of Chat Client & Models
- Essential AI Concepts for Spring AI Projects
- Using Multiple LLM Models Together
- Prompting & Response Parsing (Entities & Lists)
- Default Prompts in Spring AI
- Dynamic Prompt Templating
- Advanced Prompt Templating Techniques
- Key Considerations Before Building AI Apps
- Advisors in Spring AI
- Custom Advisor for Token Usage Tracking
- Streaming AI Responses
- Implementing LLM Memory
- Step-by-Step Chat Memory Implementation
- Multi-User Interaction with LLMs
- In-Memory Chat Memory Repository
- Introduction to RAG (Retrieval-Augmented Generation)
- Vectors, Embeddings & Vector Databases
- Storing Embeddings in Vector Database
- Similarity Search & Data Passing to LLM
- Automatic Prompt Handling in Spring AI
- Advanced RAG Architecture
- Complete RAG Pipeline Implementation
- ETL Pipeline (Extract, Transform, Load)
- Document Reader, Transformer & Writer
- Running AI Models Locally with Docker
- Tool Calling for Real-World AI Actions
- Building AI Agents with Spring AI
- AI Help Desk Project
- Help Desk System with Spring Boot + Spring AI
- Bug Fixing & Conversation Management
- Backend Completion, Streaming & DB Storage
- Full-Stack Help Desk UI (React, Tailwind, Shadcn UI)
- Audio-to-Text using Spring AI

Front End Development Module

HTML

- HTML + Basic HTML structure
- Formatting tags & meta tags
- Semantic HTML5 Elements
- List & link + inline and block tags
- Form tags with examples
- Table + image + media tags
- HTML Entities tags
- Iframes + HTML APIS
- SEO TAGS + HTML5 TAGS
- HTML5 tags

JavaScript

- What is JavaScript & Why use it
- Variable declaration using JS, closure concept
- decision making and control statement
- Array & String using JS
- Event Handling in JavaScript
- DOM (Document object model)
- Arrow function
- Rest operator & Spread operator
- Validation using JavaScript
- Promise, Async and Await
- Event loop concept
- JSON & AJAX Concept
- Fetch API
- Function currying concept

- 15 Trainer-solved assignments
- 20 Lab assignments

Next.js

- Introduction to Next.js
- Project Setup & Basic, Page Router VS App Router, App Router Fundamental
- Data Fetching using page router and App router, Styling in next.js, static site generation, Server-side rendering
- client components, API routes, Authentication, Form Validation
- Database & ORM integration, State management

CSS

- Introduction of CSS + Types of CSS
- Selectors in CSS
- Colors & Background properties
- Font & Text Styling properties
- Border, Margins and padding
- Display and Visibility properties
- Positioning and Z-index
- Flexbox
- CSS Grid Layout
- CSS units and Measurements
- Media Queries (Responsive design)
- Transition and animations
- CSS variables (Custom properties)
- Advanced CSS Feature

- 10 CSS Templates design by trainer
- 10 CSS Templates lab assignments

React + Redux + Next.js

- Introduction about React
- Creating component
- class component with state
- Life cycle implementation of class component
- CSS using React
- Bootstrap using React
- Event handling and form handling using class component
- validation using React
- Function component
- Difference between function and class component
- Hooks concept in React &its benefit
- useState, useEffect, useReducer, useMemo(), useContext(), useRef(), useCallback() etc
- form handling with function component
- Routing concept in React
- API & Async in React
- Build tools and development like as vercel, netify etc
- React with redux
- validation using React

- 20 Assi. & mini project solved by trainer
- 30 Lab assignments

Tools to Master

Tools to Master Java Full Stack

DEV TOOLs Syllabus

1. Maven + Lombok + logging

- What is maven
- Installing & setting up maven
- Understanding POM.xml
- Maven Life Cycle
- Dependency Management
- Maven plugins
- Maven project type
- Maven profile
- What is Lombok
- Boilerplate code reduction
- Builder and singular annotation
- Logging with Lombok
- What is logging
- logging using slf4j+logback

2. DOCKER + JUNIT + Mockito

DOCKER

1. Introduction to Docker

- What is Docker?
- Containers vs Virtual Machines
- Docker Architecture
- Use cases in Java development
- Dependency Management

2. Installing and Running Docker

- Install Docker (Docker Desktop)
- Docker CLI basics
- Dockerfile overview
- docker build, docker run, docker ps, docker stop

3. Dockerfile & Image Creation

- Writing a Dockerfile for a Java/Spring Boot app
- Best practices (multi-stage builds, slim images)
- .dockerignore, Exposing ports
- Setting env variables

4. Working with Containers

- Starting, stopping, deleting containers
- Volume mounting
- Logs and debugging containers
- Environment variable injection

5. Docker Compose

- Defining services with docker-compose.yml
- Connecting app with MySQL/PostgreSQL/Redis containers
- Networking between services
- Common commands (up, down, logs, exec)

6. Docker with Spring/Spring Boot

- Containerize Spring Boot app
- Use Docker Compose for full stack (Spring + DB)
- Build once, run anywhere principle
- Profiles for dev/prod in Docker

JUNIT

1. Introduction to Testing in Java

- Why testing matters
- Types of testing: unit, integration, end-to-end

2. JUnit 5 Basics

- Setting up JUnit 5 (Maven/Gradle)
- Writing your first test
- Test lifecycle
- @BeforeEach, @AfterEach
- @BeforeAll, @AfterAll
- Assertions
- assertEquals, assertTrue, assertThrows
- @DisplayName, @Disabled

3. Parameterized Tests

- Using @ParameterizedTest
- @ValueSource, @CsvSource, @MethodSource

4. Nested Tests & Tagging

- @Nested classes for logical grouping
- @Tag and filtering test groups

5. JUnit with Maven and Gradle

- Running tests from CLI
- Reports and test coverage

6. Testing Spring Boot Apps

- Using @SpringBootTest
- @WebMvcTest for controller layer
- @DataJpaTest for repository layer
- Test-specific properties (application-test.yml)

Mockito

1. Introduction to Mockito

- Why mocking is needed
- When to use mocking vs stubbing

2. Mockito Basics

- Adding Mockito to your project
- Creating mocks using @Mock
- Injecting mocks with @InjectMocks
- Basic when(...). then Return(...) usage
- Verifying interactions with verify(...)

3. Stubbing and Behavior Testing

- Return different values on successive calls
- Throwing exceptions with thenThrow
- Mocking void methods with doNothing

4. Argument Matchers

- Using any(), eq(), argThat()
- Capturing arguments with ArgumentCaptor

5. Testing Services with Mockito

- Mocking service layer dependencies
- Writing clean unit tests for business logic
- Mocking DAOs or REST clients (e.g., using RestTemplate)

6. JUnit with Maven and Gradle

- Integration with @MockBean
- Combining with @WebMvcTest, @SpringBootTest
- Testing with in-memory databases

Project Training

Transform Skills into Experience & Project Training with Mentorship

Project Training Syllabus

High level design

Module 1: Introduction to System Design

- What is System Design?
- Difference between HLD and LLD
- Role of HLD in SDLC
- Types of Systems (monolithic, distributed, microservices
- Real-world importance of system scalability and availability

Module 2: Requirements Gathering for Design

- Understanding functional and nonfunctional requirements
- Capacity estimation (users, data, traffic)
- Identifying constraints: latency, availability, consistency, etc
- Use case definition and prioritization

Module 3: Architectural Patterns & Styles

- Monolithic Architecture
- Microservices Architecture
- Event-driven Architecture
- Layered, Modular, and ClientServer Architectures
- Service-Oriented Architecture (SOA)

Module 4: High-Level System Components

- Defining major modules/services
- System workflow and request flow
- Third-party integrations
- Design of APIs at a high level

Module 5: Data Flow and Communication

- Synchronous vs Asynchronous communication
- REST vs gRPC vs Messaging Queues
- Load balancing and reverse proxies
- Data flow diagrams and component interaction

Module 6: Database and Storage Design

- High-level database selection: SQL vs NoSQL
- Horizontal scaling (sharding, partitioning)
- Caching strategies (Redis, Memcached)
- Storage architecture (Blob storage, CDN, file systems)

Module 7: Non-Functional Aspects of HLD

- Scalability (vertical & horizontal)
- Availability & fault tolerance
- Reliability & durability
- Security (authentication, authorization, data protection)
- Maintainability and extensibility

Module 8: Design Documentation and Diagrams

- UML diagrams: Component, Deployment, Activity
- Sequence and flow diagrams
- High-level architecture diagrams (draw.io, Lucidchart)
- Tech stack and component justifications

Low level system design

Module 1: Introduction to Low-Level Design

- What is Low-Level Design (LLD)?
- Difference between HLD and LLD
- EImportance of LLD in software development
- Real-world relevance: clean, scalable, maintainable code

Module 2: Object-Oriented Programming (OOP) Principles

- Encapsulation, Inheritance, Polymorphism, Abstraction
- Real-world modeling using OOP
- Composition vs Inheritance
- Access modifiers and encapsulation best practices

Module 3: Class & Interface Design

- Identifying entities and responsibilities
- Designing classes with proper attributes and methods
- Interface segregation and decoupling
- UML Class Diagrams

Module 4: Code Structuring & Modularization

- Layered architecture in code (Controller, Service, DAO, etc.)
- Folder/package structure
- Code reusability and modularization
- Handling common utilities and shared services

Module 5: Error Handling, Logging & Exception Design

- Custom exception hierarchy
- Best practices in error reporting
- Logging strategies (log levels, structured logging)

Module 6: Test-Driven Design & Code Quality

- Introduction to TDD (Test-Driven Development)
- Writing unit tests (JUnit/TestNG)
- Mockito and mocking dependencies
- Code coverage and quality metrics

Career Readiness Program

Master Aptitude, Reasoning & Soft Skills for Career Readiness

1. Quantitative Aptitude

- Number System
- Arithmetic Ability
- Percentages
- Profit, Loss, and Discount
- Simple and Compound Interest
- Ratio and Proportion
- Averages
- Time and Work
- Time, Speed, and Distance
- Algebra
- Geometry and Mensuration
- Permutation, Combination & Probability
- Data Interpretation (DI)
- Logarithms and Exponents
- Set Theory & Venn Diagrams

2. Logical Reasoning

- Coding and Decoding
- Series and Sequences
- Blood Relations
- Direction Sense Test
- Syllogisms
- Analogy
- Logical Puzzles
- Data Sufficiency
- Clocks and Calendars
- Statement and Assumption / Conclusion
- Alphanumeric Series
- Cubes and Dice

3. Soft skill

- Communication Skills
- Presentation Skills
- Group Discussion (GD) Skills
- Personal Interview Preparation
- Resume Building
- Emotional Intelligence (EQ)
- Time Management & Productivity
- Teamwork & Leadership
- Professional Etiquette
- Personality Development

Placement Preparation

Industry-Aligned Placement Preparation Program

Mock interviews, real-world assessments, and expert mentorship

Placement Preparation & 100% Placement Support!

- Mock Interviews conducted every 20 days by GIRI'S TECHHUB.
- Mock process includes:
  Programming Test | L1 Technical | L2 Technical | HR Interview

Feedback & Improvement

- Upload audio/video recordings on the GIRI'S TECHHUB App for feedback correction.
- Written feedback will be reviewed and refined for better clarity.

Mock Interview Guidelines

- Mock tests are based only on completed syllabus.
- Test dates announced 15–20 days in advance.
- Study plan & interview question bank provided beforehand.

Placement Support Policy

- 80% attendance mandatory for theory & practical sessions.
- Project submission & mock interview attendance compulsory.
- Minimum 50% in L1 interview required for placement recommendation.
- Extra mentorship & re-attempts every 15 days if L1 is not cleared.

L1 & L2 Levels

- L1: Below-average preparedness
- L2: Average to excellent preparedness

Program at Glance!!

COHORT HIGHLIGHTS

A 100% practical, coding focused cohort , designed by top industry maestros to make you job-ready.!!

Classroom & Online Java Full Stack Mentor Sessions

In classroom & online sessions, every concept is clearly explained and made easy to understand for each student. Personalized one-to-one guidance ensures strong practical knowledge, clear understanding, and continuous progress tracking...

Classroom Practical Assignment-Solving Sessions

Hands-on classroom sessions focus on solving real-time assignments under trainer supervision. Students practice coding, logic building, and implementation in a structured lab environment, ensuring a strong practical understanding of concepts.

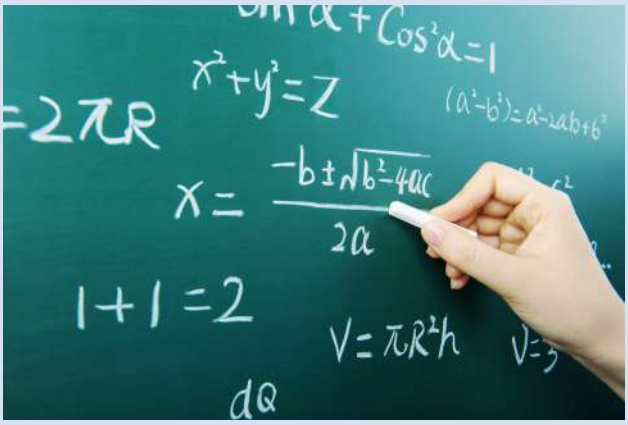
Online Practical Assignment-Solving Sessions

Live online sessions for remote or hybrid learners. Students work on assignments in real time, interact with mentors, share screens, and receive instant feedback—ensuring practical learning without compromising flexibility.

Soft Skills Sessions by Mentors

Dedicated sessions to enhance communication, presentation skills, teamwork, professional etiquette, and confidence. These sessions prepare students for HR interviews, client interactions, and corporate work culture.

Quantitative Aptitude Sessions by Mentors



Focused training on logical reasoning, numerical ability, and problem-solving techniques. Helps students clear written tests, aptitude rounds, and technical assessments in campus drives and company interviews.

Motivational Speaker Sessions by Industry Experts

Motivational sessions with industry professionals and successful alumni. Students gain real-world insights, career guidance, exposure to current industry trends, and motivation to stay focused on long-term career goals.

Monthly Syllabus-Based Mock Interviews / Machine Tests

Mock interviews and machine tests are conducted every month based on the syllabus completed so far. Industry experts evaluate students on technical knowledge, communication skills, and problem-solving ability, followed by detailed feedback for improvement.

Console-Based Projects

Focused training on logical reasoning, numerical ability, and problem-solving techniques. Helps students clear written tests, aptitude rounds, and technical assessments in campus drives and company interviews.

Live Real-Time Project Work

Every student starts working on live real-time projects. Students are grouped and assigned complete projects to work on collaboratively, gaining hands-on exposure to real-world development workflows, tools, and team collaboration.

100% Placement Support..!

Students start receiving placement interview calls. Our dedicated placement team coordinates with students, schedules interviews, provides interview guidance, and stays in continuous touch until the student is successfully placed.

Meet your Mentors!!

Mr. Adinath Giri

Java Expert | 15+ Years of IT Training Experience

(Founder & CEO of Giri's Tech Hub Pvt. Ltd. Pune)

Giri Sir holds an MCA (2010) from Government College of Engineering, Chhatrapati Sambhajinagar with expertise in Core & Advanced Java and Full Stack development.

He is proficient in the Spring Framework and modern Java technologies, building scalable applications.

He has enabled the placement of 5400+ students through 2,000+ recruitment drives. As Director of Giri's Tech Hub, led 500+ batches, and conducted 300+ college workshops.

Our Alumni Work

Our Alumni at Leading Companies

Building successful careers across top global organizations.

cognizant

SYNTEL
Consider IT Done

TCS TATA CONSULTANCY SERVICES

Capgemini

accenture

Persistent

BARCLAYS

HCL

Morgan Stanley

LTIMindtree

wipro

Infosys

TECH mahindra

Mphasis
The Next Applied

Go for it!!

Ready to Start Your Career Journey?

Industry-Aligned Training | Expert Mentorship | Placement Support

At Giri's Tech Hub, we are committed to transforming aspiring learners into job-ready professionals through practical training, real-world projects, and structured placement preparation.

Contact Us

Head Office : Office No. 102, Dangat Park, Mumbai–Bangalore Highway, Warje, Pune

Our Branch : Office No. 11, Hariom Tower, Opp. off Govt. Engineering College, Chh. Sambhaji Nagar

+91 91 75444433 | +91 9049361265

www.techhubindia.org