

HackForge

Spring boot and React Full Stack Certification Training

Total 60 Hours

Live Online | Andhra Pradesh | Telangana | Tamil Nadu | Karnataka

Phone :9884789292

Module 1: Java Fundamentals

- * History & Features of Java
- * JVM vs JRE vs JDK
- * Java Program Structure
- * Compilation & Execution Flow
- * Data Types (Primitive & Non-Primitive)
- * Variables (Local, Instance, Static)
- * Type Casting (Implicit & Explicit)
- * Operators
- * Control Statements
 - * if / else
 - * switch
 - * for loop
 - * while loop
 - * do-while loop
 - * break & continue

Module 2: Object-Oriented Programming (OOPs)

Core Concepts

- * Class & Object

- * Encapsulation
- * Inheritance
- * Polymorphism
- * Abstraction

Inheritance Types

- * Single
- * Multilevel
- * Hierarchical

Important Keywords

- * this
- * super
- * final
- * static

Module 3: Constructors

- * Default Constructor
- * Parameterized Constructor
- * Constructor Overloading
- * Constructor Chaining
- * Private Constructor

Module 4: Strings & Wrapper Classes

String Handling

- * String class
- * String immutability
- * equals() vs ==
- * hashCode()
- * String methods
- * String vs StringBuilder vs StringBuffer

Wrapper Classes

- * Primitive vs Wrapper
- * Autoboxing & Unboxing
- * Use in Collections

Module 5: Exception Handling

Concepts

- * What is Exception
- * Checked vs Unchecked Exceptions
- * Throwable hierarchy

Keywords

- * try
- * catch
- * finally
- * throw
- * throws

Custom Exceptions

- * User-defined exceptions
- * Best practices

Module 6: Interfaces & Abstract Classes

- * Interface vs Abstract class
- * Multiple inheritance using Interface
- * Functional Interface
- * Marker Interface
- * Default & static methods in interfaces

Module 7: Java Collections Framework

Collection Hierarchy

- * List
- * Set
- * Queue
- * Map

List

- * ArrayList
- * LinkedList
- * Vector
- * Stack

Set

- * HashSet
- * LinkedHashSet
- * TreeSet

Map

- * HashMap
- * LinkedHashMap
- * TreeMap
- * Hashtable

Utilities

- * Comparable vs Comparator
- * Iterator

- * Collection vs Collections

Module 8: Java 8 Features

Lambda Expressions

Functional Interfaces

- * Predicate
- * Consumer
- * Supplier
- * Function

Stream API

- * Stream creation
- * filter
- * map
- * reduce
- * forEach
- * Collectors

Other Features

- * Optional class
- * Method references
- * New Date & Time API

Module 9: Multithreading & Concurrency

- * What is Thread
- * Thread Life Cycle

Creating Threads

- * Thread class
- * Runnable interface

Synchronization

- * Thread safety
- * Deadlock
- * wait()
- * notify()
- * notifyAll()

Advanced Concepts

- * Executor framework
- * volatile keyword

Module 10: File Handling & Serialization

- * File class
- * Byte streams
- * Character streams
- * Serialization
- * Deserialization
- * transient keyword

Module 11: Java Memory Management

- * Stack vs Heap
- * JVM Memory Areas
- * Garbage Collection
- * Memory leaks
- * finalize()

Module 12: equals(), hashCode(), toString()

- * Contract rules
- * Usage in HashMap
- * Usage in HashSet
- * Overriding best practices

Module 13: Important Java Keywords

- * static
- * final
- * transient
- * volatile
- * synchronized
- * native
- * strictfp

Module 14: Design Principles & Best Practices

- * SOLID principles (core understanding)
- * Immutability
- * Defensive copying
- * Clean code practices
- * Readable class design

Module 15: JDBC Connectivity

- * JDBC Architecture
- * Driver types
- * Connection setup
- * Statement vs PreparedStatement
- * ResultSet handling
- * CRUD operations

Maven syllabus:

1. Introduction to Maven
2. Installing and Setting Up Maven
3. Maven Project Structure
4. Understanding pom.xml
5. Maven Build Lifecycle
6. Maven Commands and Execution
7. Dependency Management in Maven
8. Working with Maven Repositories
9. Maven Plugins Overview
10. Creating Maven Projects Using Archetypes
11. Building and Packaging Projects with Maven
12. Running Tests with Maven
13. Managing Project Versions
14. Integrating Maven with IDEs
15. Introduction to Maven with Spring Boot Projects

MySQL Syllabus Titles (Module-wise)

1. Introduction to Databases & MySQL Fundamentals
2. MySQL Installation & Environment Setup
3. Database and Table Creation
4. CRUD Operations in MySQL
5. Filtering, Sorting, and Conditional Queries
6. Aggregate Functions & Grouping Data
7. Constraints in MySQL
8. Joins and Relationships Between Table

Module 1: Spring Framework Fundamentals (Core Concepts)

Foundation of Spring ecosystem

- What is Spring Framework
- Advantages of Spring
- Spring vs Java EE
- Spring Architecture

IoC & Dependency Injection

- Bean lifecycle
- BeanFactory vs ApplicationContext
- @Component, @Service, @Repository
- @Autowired, @Qualifier, @Primary
- Constructor vs Setter Injection

Configuration Methods

- XML configuration
- Java configuration
- @Configuration
- @Bean
- Component scanning

Bean Scopes & Lifecycle

- Singleton
- Prototype
- Request scope
- Session scope
- @PostConstruct
- @PreDestroy

Module 2: Spring Boot Fundamentals

Rapid backend application development

- Starter dependencies
- Auto-configuration
- Embedded Tomcat
- application.properties
- application.yml
- Profiles & environment configs

Module 3: Spring MVC (REST API Development)

Building production-ready APIs

MVC Architecture

- DispatcherServlet
- Controller layer
- Service layer
- Repository layer

REST Controller Annotations

- @Controller vs @RestController
- @RequestMapping
- @GetMapping / @PostMapping
- @PutMapping / @DeleteMapping
- @RequestParam
- @PathVariable
- @RequestBody
- @ResponseBody

REST Best Practices

- HTTP methods
- Status codes
- Pagination & sorting
- DTO pattern

Module 4: Validation & Exception Handling

Robust API error handling

- Bean Validation (JSR-380)
 - @Valid usage
 - @ControllerAdvice
 - @ExceptionHandler
 - Custom error response structure
 - Global exception handling
-

Module 5: Spring Data JPA & Hibernate

Database integration layer

JPA Basics

- Entity mapping
- @Entity
- @Table
- @Id
- @GeneratedValue

Relationships

- OneToOne
- OneToMany
- ManyToOne
- ManyToMany

Repository Layer

- CrudRepository
 - JpaRepository
 - Custom queries
 - JPQL
 - Native queries
-

Module 6: Spring AOP (Aspect-Oriented Programming)

Cross-cutting concerns

AOP Concepts

- Aspect
- Join Point
- Pointcut
- Advice types

Implementation

- @Aspect
- @EnableAspectJAutoProxy
- Execution expressions

Real Use Cases

- Logging
 - Transactions
 - Performance monitoring
 - Security enforcement
-

Module 7: Spring Security + JWT Authentication

Enterprise backend security

Security Basics

- Authentication vs Authorization
- Security filter chain

Spring Security Setup

- In-memory authentication
- Database authentication
- UserDetailsService
- BCrypt password encoding

JWT Authentication

- Token structure
- Token generation
- Token validation
- Stateless authentication
- Refresh tokens

Role-Based Access

- @PreAuthorize
 - @RolesAllowed
-

Module 8: Unit & Integration Testing (JUnit + Mockito)

Production-quality backend testing

JUnit 5

- Test lifecycle annotations
- Assertions
- Parameterized tests

Mockito

- Mocking dependencies
- @Mock
- @InjectMocks
- Verify interactions

Spring Boot Testing

- @SpringBootTest
 - @WebMvcTest
 - @DataJpaTest
-

Module 9: Microservices Architecture

Distributed backend systems

Core Concepts

- Monolith vs Microservices
- Service boundaries
- Database per service
- Inter-service communication

Spring Cloud Components

- Config Server
- Eureka Service Discovery
- Load balancing
- Resilience4j Circuit Breaker

Communication Patterns

- REST communication
 - Feign Client
 - Sync vs Async calls
-

Module 10: API Gateway

Centralized request routing

Gateway Features

- Routing
- Filtering
- Rate limiting
- Authentication centralization

Spring Cloud Gateway

- Route predicates
 - Global filters
 - Custom filters
 - JWT validation at gateway
-

Module 11: Docker & CI/CD (GitHub Actions)

DevOps for backend engineers

Docker

- Docker architecture
- Dockerfile
- Images vs Containers
- Multi-stage builds
- Docker Compose

GitHub Actions

- CI workflow setup
 - Build pipeline automation
 - Test execution pipeline
 - Docker image build
 - Push to Docker Hub / ECR
-

Module 12: AWS Deployment

Production deployment strategies ☁️

Core AWS Services

- EC2
- S3
- RDS
- IAM basics

Deployment Options

- EC2 + Docker deployment
- Elastic Beanstalk
- ECS (overview)

AWS Best Practices

- Environment variables
- Secrets management
- Monitoring
- Logging
-

React Certification Training

20 Hours

Module 1: Web Development Fundamentals

- How the Internet works
 - Client vs Server
 - HTTP / HTTPS basics
 - Browser rendering process
 - Developer tools (Chrome DevTools)
 - VS Code setup & extensions
 - Git & GitHub basics
-

Module 2: HTML5 Essentials

Structure of modern web pages

- HTML document structure
- Semantic elements
- Headings, paragraphs, links
- Lists & tables
- Images, audio, video
- Forms & input types

- Accessibility basics
-

Module 3: CSS3 Fundamentals

Designing responsive layouts

- Selectors & specificity
 - Colors & typography
 - Box model
 - Display properties
 - Positioning
 - Flexbox
 - Grid layout
 - Media queries
 - Responsive design
-

Module 4: Advanced CSS & UI Design

Modern interface styling

- CSS variables
 - Transitions & animations
 - Hover effects
 - Responsive navigation bars
 - Card layouts
 - Mobile-first design
 - Tailwind CSS (optional)
-

Module 5: JavaScript Fundamentals

Core programming logic for frontend

- Variables (var, let, const)
- Data types
- Operators
- Functions
- Arrays & objects
- Loops
- Conditional statements
- DOM manipulation
- Event handling

Module 6: Advanced JavaScript (ES6+) ★★☆☆

Modern JavaScript required for React

- Arrow functions
- Destructuring
- Spread & rest operators
- Template literals
- Modules (import/export)
- Closures
- Hoisting
- Callbacks
- Promises
- Async / Await
- Fetch API

Module 7: React Fundamentals ★★☆☆

Component-based UI development

- Introduction to React
- JSX
- Functional components
- Props
- State
- Event handling
- Conditional rendering
- Lists & keys

Module 8: React Hooks (Modern React)

Core hooks used in real projects

- useState
 - useEffect
 - useRef
 - useContext
-

Module 9: React Advanced Concepts

Building scalable applications

- Component architecture
 - Reusable components
 - Forms handling
 - Lifting state up
 - Context API
 - Custom hooks
 - Environment variables
-

Module 10: Routing & API Integration

Connecting frontend with backend

- React Router setup
 - Dynamic routes
 - Navigation handling
 - Axios integration
 - Fetching REST APIs
 - Loading states
 - Error handling
-

Module 11: Project Development

Real-world frontend application

- Authentication UI
 - Dashboard layout
 - API-based project
 - Responsive UI implementation
 - Component reuse strategy
-

Module 12: Deployment & Best Practices

Making apps production-ready

- Build optimization
- Environment configs
- Deploy to Vercel / Netlify

- Performance improvements
- Clean folder structure

Git & Version Control

- Version control concepts
- Git installation
- Git workflow
- Branching strategies
- Merge vs rebase
- Resolving conflicts
- GitHub / GitLab usage

Commands:

git clone
git commit
git push
git pull
git branch
git merge

Module 1: DevOps Fundamentals

- What is DevOps?
- DevOps lifecycle
- Agile vs DevOps
- CI/CD concepts
- Infrastructure as Code (IaC)
- DevOps roles & responsibilities
- DevOps toolchain overview

Module 4: CI/CD with Jenkins

- Continuous Integration basics
- Jenkins architecture
- Installing Jenkins
- Creating pipelines
- Freestyle jobs
- Jenkinsfile (pipeline as code)
- Integrating Git with Jenkins
- Automated builds

Module 5: Docker (Containerization)

- Virtualization vs Containers
- Docker architecture
- Installing Docker
- Docker images & containers
- Dockerfile creation
- Docker volumes
- Docker networking
- Docker Compose

Cloud Computing (AWS Recommended)

Core services:

- EC2
- S3
- IAM
- VPC basics
- Load Balancer
- Auto Scaling
- CloudWatch monitoring

Deployment practice:

- Host application on EC2
- Store files in S3
- Configure IAM roles

Monitoring & Logging

Tools:

- Prometheus
- Grafana
- ELK Stack

Concepts:

- Metrics monitoring
- Log aggregation
- Alerting systems
- Dashboard creation

Social Media Links:

<https://www.linkedin.com/company/hackforge-academy>

https://www.instagram.com/hackforge_english/

https://www.youtube.com/@hackforge_code

**Daily coding whatsapp community in java,React,JavaScript,Python
+919884789292**