

# HackForge

## Core Java Certification Training

40 Hours

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

Phone :9884789292

### 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
- 

## MySQL Syllabus

### Module 1: Introduction to Databases

- What is DBMS
- What is RDBMS
- Features of MySQL
- Installing MySQL Server & Workbench
- Creating database
- Selecting database

Commands:

```
CREATE DATABASE academy;  
USE academy;  
SHOW DATABASES;
```

---

## Module 2: Data Types in MySQL

- Numeric types (INT, BIGINT, FLOAT, DECIMAL)
  - String types (CHAR, VARCHAR, TEXT)
  - Date & time types (DATE, DATETIME, TIMESTAMP)
  - Boolean type
  - ENUM & SET (optional)
- 

## Module 3: Table Creation & Structure

- CREATE TABLE
- DESCRIBE TABLE
- ALTER TABLE
- DROP TABLE
- TRUNCATE TABLE
- Rename table

Example:

```
CREATE TABLE students (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  name VARCHAR(100),  
  email VARCHAR(100),  
  age INT  
);
```

---

## Module 4: CRUD Operations (Core SQL)

### INSERT

```
INSERT INTO students(name, age)  
VALUES ('Praveen', 22);
```

### SELECT

```
SELECT * FROM students;
```

### UPDATE

```
UPDATE students  
SET age = 23  
WHERE id = 1;
```

## DELETE

```
DELETE FROM students  
WHERE id = 1;
```

---

## Module 5: Constraints

- PRIMARY KEY
- FOREIGN KEY
- UNIQUE
- NOT NULL
- DEFAULT
- CHECK (concept)

Example:

```
email VARCHAR(100) UNIQUE
```

---

## Module 6: Filtering Data

- WHERE clause
- AND, OR, NOT
- BETWEEN
- IN
- LIKE
- IS NULL

Example:

```
SELECT * FROM students  
WHERE age BETWEEN 18 AND 25;
```

---

## Module 7: Sorting & Limiting Data

- ORDER BY ASC / DESC
- LIMIT
- DISTINCT

Example:

```
SELECT * FROM students  
ORDER BY age DESC  
LIMIT 5;
```

---

## Module 8: Aggregate Functions

- COUNT()
- SUM()
- AVG()
- MIN()
- MAX()

Example:

```
SELECT COUNT(*) FROM students;
```

---

## Module 9: GROUP BY & HAVING

- GROUP BY usage
- HAVING vs WHERE difference

Example:

```
SELECT age, COUNT(*)  
FROM students  
GROUP BY age;
```

---

## Module 10: Joins (Very Important)

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- SELF JOIN (optional)

Example:

```
SELECT students.name, courses.title  
FROM students  
INNER JOIN courses  
ON students.course_id = courses.id;
```

---

## Module 11: Keys & Relationships

- Primary key concept
- Foreign key relationships
- One-to-One
- One-to-Many

- Many-to-Many

Example:

```
FOREIGN KEY (course_id)  
REFERENCES courses(id)
```

---

## Module 12: Indexes

- What is indexing
- Creating index
- Unique index
- Composite index
- Performance benefits

Example:

```
CREATE INDEX idx_email  
ON students(email);
```

---

## Module 13: Views

- What is a view
- Creating view
- Updating view
- Dropping view

Example:

```
CREATE VIEW active_students AS  
SELECT * FROM students  
WHERE status = 'active';
```

---

## Module 14: Stored Procedures

- Creating stored procedure
- Calling procedure
- Parameters (IN, OUT)

Example:

```
CREATE PROCEDURE getStudents()  
SELECT * FROM students;
```

---

## Module 15: Functions

- Creating functions
- Returning values

Example:

```
CREATE FUNCTION total_students()  
RETURNS INT  
RETURN (SELECT COUNT(*) FROM students);
```

---

## Module 16: Transactions

- START TRANSACTION
- COMMIT
- ROLLBACK
- SAVEPOINT
- ACID properties

Example:

```
START TRANSACTION;  
UPDATE accounts SET balance = balance - 500 WHERE id = 1;  
COMMIT;
```

---

## Module 17: Subqueries

- Single-row subquery
- Multi-row subquery
- Nested queries

Example:

```
SELECT name  
FROM students  
WHERE age > (SELECT AVG(age) FROM students);
```

---

## Module 18: Advanced Queries

- EXISTS
- ANY
- ALL
- CASE statement

## **Social Media Links:**

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

[https://www.instagram.com/hackforge\\_english/](https://www.instagram.com/hackforge_english/)

[https://www.youtube.com/@hackforge\\_code](https://www.youtube.com/@hackforge_code)

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

www.hackforge.in