**Comprehensive Spring Boot 15-Day Schedule**

**Day 1: Core Java Basics**

* Variables, Data Types, Operators, and Control Statements
* Arrays and Strings
* Object-Oriented Programming (OOP): Classes, Objects, Inheritance, Polymorphism, Abstraction, Encapsulation

**Day 2: Advanced Java**

* Collections Framework: List, Set, Map, Queue
* Generics and Enumerations
* Exception Handling

**Day 3: Multithreading and Java 8+ Features**

* Multithreading and Concurrency: Threads, Synchronization, Executors
* Java 8+ Features: Stream API, Lambda Expressions, Optional, Date/Time API
* Input/Output Streams: File Handling

**Day 4: Build Tools and Spring Core**

* Build Tools: Maven or Gradle (Dependency Management, POM/Build.gradle)
* Introduction to Dependency Injection (DI) and Inversion of Control (IoC)
* ApplicationContext vs. BeanFactory
* Configuring Beans (@Component, @Service, @Repository, @Controller)

**Day 5: Spring Core (Continued)**

* Constructor Injection, Setter Injection
* Bean Lifecycle (@PostConstruct, @PreDestroy)
* Bean Scopes (Singleton, Prototype, etc.)

**Day 6: Spring AOP and MVC**

* What is AOP?
* Cross-cutting concerns like logging, security
* Pointcuts, Advices (@Before, @After, @Around)
* Defining Aspects (@Aspect Annotation)
* DispatcherServlet and Request Lifecycle
* @Controller vs. @RestController

**Day 7: Spring MVC (Continued) and Spring Boot Fundamentals**

* Model-View-Controller Architecture
* View Resolvers (JSP, Thymeleaf)
* What is Spring Boot, and how is it different from Spring Framework?
* Auto-Configuration and Starter Dependencies
* Creating a Spring Boot Project with Spring Initializr

**Day 8: RESTful API Development**

* REST principles: Statelessness, Resource Representations
* HTTP Methods: GET, POST, PUT, DELETE, PATCH
* @RequestMapping, @GetMapping, @PostMapping, etc.
* Path Variables and Query Parameters
* Using @RequestBody and @ResponseBody

**Day 9: Validation, Exception Handling, and Spring Data JPA**

* Validating Requests using @Valid and @Validated
* Custom Validators (@Constraint)
* Global Exception Handling with @ControllerAdvice
* Defining Custom Exceptions
* What is JPA and ORM?
* Configuring a DataSource (H2, MySQL, PostgreSQL)

**Day 10: Spring Data JPA (Continued)**

* Defining Entity Classes (@Entity, @Table)
* Creating Repositories (JpaRepository, CrudRepository)
* Writing JPQL and Native Queries
* Working with Pagination and Sorting

**Day 11: Advanced JPA and Hibernate**

* Entity Relationships: One-to-One, One-to-Many, Many-to-Many
* Cascade Types (PERSIST, MERGE, REMOVE, ALL)
* Fetch Types: Lazy vs. Eager Loading
* First-Level and Second-Level Caching

**Day 12: Spring Security and JWT**

* Setting Up Spring Security
* Authentication and Authorization Basics
* Password Encoding with BCrypt
* Role-Based Access Control
* What is JWT and its Structure?
* Generating and Validating JWT Tokens
* Adding JWT to Secured REST APIs

**Day 13: Microservices and Messaging**

* Understanding Microservices Architecture
* Service Discovery with Eureka
* API Gateway with Spring Cloud Gateway
* Inter-Service Communication (Feign Client, RestTemplate, WebClient)
* RabbitMQ: Setting up Queues, Exchanges, and Bindings
* Kafka: Producers, Consumers, and Topics

**Day 14: Caching, Asynchronous Processing, and Monitoring**

* Basics of Caching: What is Caching?
* Cache Providers: Redis, Ehcache
* Using @Cacheable, @CacheEvict, and @CachePut
* @Async and Configuring Executors
* CompletableFuture and Async Operations
* Spring Boot Actuator Basics
* Integrating Prometheus and Grafana for Monitoring

**Day 15: Testing, Docker, and Final Enhancements**

* Unit Testing with JUnit 5
* Mocking with Mockito
* Integration Testing with @SpringBootTest
* Testing REST APIs with MockMvc
* Introduction to Docker and Containerization
* Creating Docker Images for Spring Boot Apps
* Documentation with Swagger/OpenAPI
* Building a Full-Featured Application:
  + REST APIs with CRUD Operations
  + Secure APIs with JWT
  + Use of Database Relationships
  + Implementation of Caching and Messaging
  + Deployment and Monitoring