3-Month Java Spring Boot Learning Roadmap

From Zero to Spring Boot Developer

Month 1: Java Fundamentals & Programming Basics

Week 1: Programming Foundations

Goals: Understand basic programming concepts and set up development environment

Learning Topics:

- What is programming and how computers work
- Install Java Development Kit (JDK 17 or later)
- Set up IntelliJ IDEA Community Edition or Eclipse
- Basic command line usage
- Your first "Hello World" program

Practice Projects:

- Write a simple calculator that adds two numbers
- Create a program that asks for your name and greets you

Daily Time: 2-3 hours **Resources:** Oracle Java tutorials, YouTube Java basics videos

Week 2: Java Syntax & Data Types

Goals: Master basic Java syntax and understand different data types

Learning Topics:

- Variables and data types (int, double, String, boolean)
- Basic operators (+, -, *, /, %)
- Input/output with Scanner class
- Comments and code documentation
- Basic debugging techniques

Practice Projects:

- Build a simple grade calculator
- Create a temperature converter (Celsius to Fahrenheit)
- Make a basic tip calculator

Daily Time: 2-3 hours

Week 3: Control Structures

Goals: Learn how to control program flow

Learning Topics:

- If-else statements and nested conditions
- Switch statements
- For loops, while loops, do-while loops
- Break and continue statements
- Logical operators (&&, ||, !)

Practice Projects:

- Create a number guessing game
- Build a simple menu-driven calculator
- Make a program that prints multiplication tables

Daily Time: 2-3 hours

Week 4: Methods & Object-Oriented Programming Basics

Goals: Understand methods and introduction to OOP

Learning Topics:

- Creating and calling methods
- Method parameters and return types
- Method overloading
- Introduction to classes and objects
- Instance variables and methods
- Constructors

Practice Projects:

- Create a Student class with properties and methods
- Build a simple Bank Account class
- Make a basic inventory system for a store

Daily Time: 3-4 hours

Month 2: Advanced Java & Web Development Prep

Week 5: Object-Oriented Programming Deep Dive

Goals: Master core OOP concepts

Learning Topics:

- Encapsulation (private variables, getter/setter methods)
- Inheritance and the "extends" keyword
- Polymorphism and method overriding
- Abstract classes and interfaces
- The "super" and "this" keywords

Practice Projects:

- Create an animal hierarchy (Animal -> Dog, Cat)
- Build a shape calculator with different shapes
- Design a simple employee management system

Daily Time: 3-4 hours

Week 6: Collections & Exception Handling

Goals: Learn to work with data collections and handle errors

Learning Topics:

- Arrays and ArrayList
- HashMap and LinkedList
- For-each loops
- Try-catch-finally blocks
- Common exception types
- Creating custom exceptions

Practice Projects:

- Build a contact book using ArrayList
- Create a word frequency counter using HashMap
- Make a robust calculator with error handling

Daily Time: 3-4 hours

Week 7: File I/O & Advanced Topics

Goals: Learn to work with files and understand advanced concepts

Learning Topics:

- Reading and writing files
- Working with CSV files
- Basic understanding of packages
- Static methods and variables
- Introduction to generics
- Basic threading concepts

Practice Projects:

- Create a simple text-based database
- Build a log file analyzer
- Make a basic file backup utility

Daily Time: 3-4 hours

Week 8: Web Development Fundamentals

Goals: Understand web development basics before Spring Boot

Learning Topics:

- How the web works (HTTP, requests, responses)
- HTML basics (forms, tables, links)
- CSS fundamentals for styling
- JavaScript basics (variables, functions, DOM)
- JSON format and parsing
- REST API concepts

Practice Projects:

- Create a simple HTML portfolio page
- Build a basic web form
- Make a simple interactive web page with JavaScript

Daily Time: 3-4 hours

Month 3: Spring Boot & Web Applications

Week 9: Spring Boot Introduction

Goals: Understand Spring Framework and create your first Spring Boot app

Learning Topics:

- What is Spring Framework and Spring Boot
- Dependency Injection and Inversion of Control
- Maven basics and dependencies
- Creating your first Spring Boot project using Spring Initializr
- Understanding application.properties
- Spring Boot annotations (@SpringBootApplication, @RestController)

Practice Projects:

- Create a "Hello World" REST API
- Build a simple calculator API
- Make a basic student information API

Daily Time: 4-5 hours

Week 10: REST APIs & Web Controllers

Goals: Build RESTful web services

Learning Topics:

- @RestController and @RequestMapping
- HTTP methods (GET, POST, PUT, DELETE)
- @PathVariable and @RequestParam
- @RequestBody and @ResponseBody
- HTTP status codes
- Testing APIs with Postman

Practice Projects:

- Create a book management API
- Build a simple blog post API
- Make a user registration API

Daily Time: 4-5 hours

Week 11: Data Persistence & Databases

Goals: Learn to work with databases in Spring Boot

Learning Topics:

- Introduction to databases and SQL

- H2 in-memory database
- Spring Data JPA basics
- Entity classes and @Entity annotation
- Repository pattern and JpaRepository
- Basic CRUD operations

Practice Projects:

- Create a task management system with database
- Build a simple e-commerce product catalog
- Make a customer management system

Daily Time: 4-5 hours

Week 12: Full Stack Application & Deployment

Goals: Create a complete web application and deploy it

Learning Topics:

- Thymeleaf templating (or connecting to a simple frontend)
- Form handling and validation
- Error handling and custom error pages
- Application profiles (dev, prod)
- Packaging and deployment basics
- Introduction to Docker (optional)

Final Projects:

- Build a complete To-Do List web application
- Create a simple blogging platform
- Make a basic inventory management system

Daily Time: 5-6 hours

Weekly Schedule Recommendation

Monday-Wednesday-Friday: New concepts and theory (1.5-2 hours each day) **Tuesday-Thursday:** Hands-on practice and coding (2-3 hours each day) **Saturday:** Project work and review (3-4 hours) **Sunday:** Rest or light review of the week's concepts (1 hour)

Essential Tools & Resources

Development Tools:

- **IDE:** IntelliJ IDEA Community Edition (recommended) or Eclipse
- **Java:** OpenJDK 17 or later
- **Build Tool:** Maven (integrated with IDE)
- **API Testing:** Postman
- **Database:** H2 (built-in) initially, then PostgreSQL or MySQL

Learning Resources:

- **Official Docs:** Spring Boot Documentation, Oracle Java Tutorials
- **Books:** "Spring Boot in Action" by Craig Walls
- **Online Courses:** Udemy, Coursera, or free YouTube tutorials
- **Practice:** LeetCode for Java fundamentals, GitHub for project hosting

Key Websites:

- Spring Initializr (start.spring.io) for project creation
- Stack Overflow for problem-solving
- GitHub for version control and portfolio

Success Tips

1. **Practice Daily:** Consistency is more important than long study sessions
2. **Build Projects:** Apply what you learn immediately through projects
3. **Join Communities:** Reddit r/learnjava, Stack Overflow, Discord coding servers
4. **Version Control:** Learn Git and GitHub early, use them for all projects
5. **Don't Rush:** If a concept isn't clear, spend extra time on it
6. **Debug Often:** Learning to debug is as important as learning to code
7. **Read Error Messages:** They're your friends, not enemies
8. **Ask Questions:** Don't hesitate to seek help when stuck

Final Project Ideas

By the end of 3 months, you should be able to build:

- A REST API for a library management system
- A simple blog with user authentication
- A task management web application
- A basic e-commerce product catalog

- A customer relationship management (CRM) system

Next Steps After 3 Months

- Learn advanced Spring features (Security, Cloud)
- Study microservices architecture
- Learn a frontend framework (React, Angular, or Vue.js)
- Explore databases more deeply (PostgreSQL, MongoDB)
- Study DevOps basics (Docker, Kubernetes)
- Contribute to open-source projects

Remember: This is an intensive 3-month program. Adjust the pace based on your learning speed and available time. The key is consistent practice and building real projects!