**Week 1: Strengthen Core Java Skills**

**Day 1-2: OOP Concepts and Advanced Java**

* Revise OOP principles: inheritance, encapsulation, polymorphism, abstraction.
* Practice implementing:
  + Abstract classes and interfaces.
  + Design patterns: Singleton, Factory, Observer.
* **Tools**: IntelliJ IDEA or Eclipse.
* **Task**: Build a simple library management system using Java.

**Day 3-4: Data Structures and Algorithms (DSA)**

* Practice:
  + Arrays, Linked Lists, Stacks, Queues.
  + Sorting algorithms (merge sort, quick sort).
  + Searching (binary search, BFS/DFS).
* **Resources**: LeetCode, HackerRank.
* **Task**: Solve 5-10 problems daily.

**Day 5-6: Collections Framework**

* Master:
  + Lists, Sets, Maps.
  + ConcurrentHashMap and Thread-safe collections.
* Practice iterating, filtering, and performing operations on collections.
* **Task**: Write a program that counts the frequency of words in a document.

**Day 7: Exception Handling and File I/O**

* Learn to handle:
  + Checked vs. unchecked exceptions.
  + Custom exceptions.
* Practice file read/write operations.
* **Task**: Build a program to log user activity to a file.

**Week 2: Advanced Java and Frameworks**

**Day 8-9: Multithreading and Concurrency**

* Learn:
  + Threads and Runnable interface.
  + Synchronization, Locks, ExecutorService.
* **Task**: Build a multithreaded program that simulates a banking system.

**Day 10-11: JDBC and Database Integration**

* Connect Java with databases using JDBC.
* CRUD operations with MySQL/PostgreSQL.
* **Task**: Create a small employee database management system.

**Day 12-13: Spring Boot Basics**

* Learn:
  + Dependency Injection (DI) and Inversion of Control (IoC).
  + REST APIs with Spring Boot.
* **Task**: Build a simple REST API for an e-commerce application.

**Day 14: Testing**

* Write unit tests with JUnit.
* Learn mocking with Mockito.
* **Task**: Write test cases for the Spring Boot application.

**Week 3: Problem-Solving and Project Work**

**Day 15-16: DSA Practice**

* Focus on dynamic programming (knapsack, Fibonacci).
* Practice coding problems from past job interviews.
* **Task**: Solve 10 medium-level problems on LeetCode.

**Day 17-18: RESTful APIs and JSON**

* Create and consume RESTful APIs.
* Use JSON for data exchange.
* **Task**: Integrate your API with a front-end tool like Postman.

**Day 19-20: Git and Version Control**

* Learn:
  + Git basics (add, commit, push, pull, branches).
  + Collaborative workflows with GitHub.
* **Task**: Push your projects to GitHub with proper commits.

**Day 21: Deploy Your Application**

* Learn to deploy your Spring Boot app to:
  + **Heroku** or **AWS EC2**.
* **Task**: Host your REST API online and test it.

**Week 4: Soft Skills and Job Preparation**

**Day 22-23: Resume and LinkedIn**

* Build a professional resume:
  + Highlight your Java, Spring Boot, and project experience.
  + Quantify achievements (e.g., "Optimized code to improve app performance by 30%").
* Optimize your LinkedIn profile for recruiters.

**Day 24-25: System Design Basics**

* Learn:
  + Designing scalable systems.
  + Database sharding, caching, load balancing.
* **Task**: Design the architecture for a URL shortener or e-commerce site.

**Day 26-27: Mock Interviews**

* Participate in mock coding and behavioral interviews.
* Focus on:
  + Explaining your thought process clearly.
  + Structuring your solutions logically.

**Day 28: Final Project**

* Build a **portfolio project**:
  + E.g., "A Movie Booking System" with:
    - Java backend (Spring Boot).
    - REST APIs.
    - Database integration (MySQL).
    - Deployed on Heroku or AWS.
* Push it to GitHub.

**Day 29: Apply for Jobs**

* Start applying to jobs on:
  + LinkedIn, Glassdoor, AngelList, and local job boards.
* Customize your resume and cover letter for each role.

**Day 30: Prepare for Behavioral Questions**

* Practice answering:
  + "Tell me about yourself."
  + "What are your strengths/weaknesses?"
  + "Why do you want to work with us?"
* Use the **STAR method** (Situation, Task, Action, Result) for examples.