🗓️ Phase 1 (Weeks 1–2): Core Java & Java 8

🧩 Goal: Strong foundation in Java syntax, OOP, collections, exceptions, multithreading, and Java 8 features.

📅 Week 1 — Core Java Basics

Day 1 – Java Foundations

- **Concepts:** JVM, JRE, JDK, Data Types, Variables, Operators, Control Structures.
- **Practice:** Write simple programs using loops, conditions, and type casting.
- **Notes:** Understand how bytecode works and JIT compilation.

Day 2 – OOP (Part 1)

- **Concepts:** Classes, Objects, Encapsulation, Constructors, Access Modifiers.
- **Practice:** Create a `BankAccount` class with methods like deposit/withdraw.
- **Notes:** Learn difference between stack and heap memory.

Day 3 – OOP (Part 2)

- **Concepts:** Inheritance, Polymorphism, Overloading, Overriding.
- **Practice:** Extend `BankAccount` to `SavingsAccount` and `CurrentAccount`.
- **Notes:** Use `super` and `this` correctly.

Day 4 – Interfaces & Abstract Classes

- **Concepts:** Interface, Abstract class, Default methods.
- **Practice:** Implement multiple interfaces in a class.
- **Notes:** Understand interface evolution post-Java 8.

Day 5 – Exception Handling

- **Concepts:** try-catch-finally, Checked vs Unchecked, Custom exceptions.

- **Practice:** Build a file reader that throws custom exceptions.
- **Notes:** Learn exception hierarchy and propagation.

Day 6 – Java Collections (Part 1)

- **Concepts:** List, Set, Map, Queue; ArrayList vs LinkedList vs HashMap.
- **Practice:** Build a student record system using Map.
- **Notes:** Understand `equals()` and `hashCode()` properly.

Day 7 – Collections (Part 2)

- **Concepts:** Comparator, Comparable, Sorting, Iterators, Streams intro.
- **Practice:** Sort employees by name and salary using Comparator.
- **Notes:** Learn internal working of HashMap.

📅 Week 2 — Multithreading, Streams, and Java 8

Day 8 – Multithreading Basics

- **Concepts:** Thread class, Runnable, Thread lifecycle, Synchronization.
- **Practice:** Create a multi-threaded counter.
- **Notes:** Difference between process and thread.

Day 9 – Concurrency Utilities

- **Concepts:** ExecutorService, Future, Callable, Locks, Atomic variables.
- **Practice:** Create a thread pool to perform parallel downloads.
- **Notes:** Learn difference between synchronized block and Lock.

Day 10 – Functional Interfaces

- **Concepts:** Predicate, Function, Consumer, Supplier.
- **Practice:** Use these interfaces with lambdas.

- **Notes:** Understand SAM (Single Abstract Method) rule.

Day 11 – Lambda Expressions

- **Concepts:** Syntax, Scope, Capturing variables, Stream filter/map usage.
- **Practice:** Convert an imperative list filter to lambda form.
- **Notes:** Use effectively final variables.

Day 12 – Streams API

- **Concepts:** Stream creation, map/filter/reduce, collectors, parallel streams.
- **Practice:** Calculate average salary using Stream.
- **Notes:** Lazy evaluation in Streams.

Day 13 – Optional & Date/Time API

- **Concepts:** Optional class, `LocalDate`, `LocalTime`, `Period`, `Duration`.
- **Practice:** Build a birthdate calculator.
- **Notes:** Avoid `NullPointerException` using `Optional`.

Day 14 – Revision & Mini Project

- **Task:** Create a `Library Management System` using Streams, Collections, and Lambdas.
- **Goal:** Reinforce all Java 8 features.

🗓️ Phase 2 (Weeks 3–4): Spring & Spring Boot Fundamentals

📅 Week 3 — Spring Core Concepts

Day 1 – Spring Introduction

- **Concepts:** Dependency Injection, Inversion of Control, Beans.
- **Practice:** Create a simple Java-based Spring configuration.

Day 2 – Spring Bean Lifecycle

- **Concepts:** Bean scopes, init/destroy methods, @Component, @Bean.
- **Practice:** Build a Config class with multiple bean definitions.

Day 3 – Dependency Injection Types

- **Concepts:** Constructor, Setter, Field injection.
- **Practice:** Inject services using `@Autowired`.

Day 4 – Spring Configuration

- **Concepts:** XML vs Java Config, @Configuration, @ComponentScan.
- **Practice:** Use annotation-based config only.

Day 5 – Spring Boot Introduction

- **Concepts:** Auto-configuration, Starters, SpringApplication.run.
- **Practice:** Build a simple REST API with `@RestController`.

Day 6 – Spring Boot REST

- **Concepts:** RequestMapping, Get/Post/Delete/Put, RequestBody.
- **Practice:** Create CRUD endpoints for `Employee` entity.

Day 7 – Exception Handling

- **Concepts:** @ControllerAdvice, @ExceptionHandler, ResponseEntity.
- **Practice:** Centralize error handling in REST API.

📅 Week 4 — JPA, Validation, Testing

Day 8 – JPA & Hibernate

- **Concepts:** Entity, Repository, Persistence context.
- **Practice:** Create a `Product` entity with CRUD.

Day 9 – Relationships

- **Concepts:** OneToOne, OneToMany, ManyToMany, Cascade.
- **Practice:** Build a simple `Order`–`Item` relation.

Day 10 – Validation

- **Concepts:** @Valid, @NotNull, @Size, Custom validators.
- **Practice:** Add validation to your REST endpoints.

Day 11 – Profiles & Properties

- **Concepts:** application.yml, environment variables, @Value.
- **Practice:** Create `dev` and `prod` profiles.

Day 12 – Actuator & Logging

- **Concepts:** Spring Boot Actuator, custom endpoints, Logback.
- **Practice:** Expose health & metrics endpoints.

Day 13 – Testing

- **Concepts:** JUnit5, Mockito, Integration testing, @SpringBootTest.
- **Practice:** Test your REST APIs with MockMvc.

Day 14 – Mini Project

- **Task:** “Employee Management Service” with REST + JPA + Validation.

🗓️ Phase 3 (Weeks 5–6): Advanced Spring Boot + Integration

📅 Week 5 — Microservices, Messaging, and Caching

Day 1 – Microservices Basics

- **Concepts:** Service registry, Config server, API Gateway overview.
- **Practice:** Setup simple microservice pair.

Day 2 – Feign Client & RestTemplate

- **Concepts:** Inter-service communication, timeouts, retries.
- **Practice:** Make inter-service call between two apps.

Day 3 – Kafka / RabbitMQ

- **Concepts:** Producer/Consumer, Topic, Partition.
- **Practice:** Publish event on user registration.

Day 4 – Redis Caching

- **Concepts:** @Cacheable, @CacheEvict, cache invalidation.
- **Practice:** Cache DB queries.

Day 5 – Async & Scheduling

- **Concepts:** @Async, @Scheduled, TaskExecutor.
- **Practice:** Schedule a report generation job.

Day 6 – Security

- **Concepts:** Basic Auth, JWT, PasswordEncoder.
- **Practice:** Secure endpoints with Spring Security.

Day 7 – Observability

- **Concepts:** Micrometer, Prometheus, Zipkin tracing.
- **Practice:** Expose custom metrics.

📅 Week 6 — Docker, CI/CD, and Final Project

Day 8 – Dockerizing Apps

- **Concepts:** Dockerfile, docker-compose, container networking.
- **Practice:** Dockerize your Spring Boot app.

Day 9 – API Gateway

- **Concepts:** Routing, load balancing, rate limiting.
- **Practice:** Use Spring Cloud Gateway.

Day 10 – Fault Tolerance

- **Concepts:** Circuit Breaker, Resilience4J, Retry patterns.
- **Practice:** Add retry and fallback logic.

Day 11 – Configuration Management

- **Concepts:** Spring Cloud Config, centralized config.
- **Practice:** Externalize app properties.

Day 12 – Deployment

- **Concepts:** CI/CD basics, GitHub Actions, Jenkins pipeline.
- **Practice:** Create automated build & deploy pipeline.

Day 13 – Review & Optimization

- **Concepts:** Common pitfalls, performance tuning.
- **Practice:** Profile and tune API response time.

Day 14 – Capstone Project

- **Task:** Build “Loyalty Rewards Service” or “File Upload Service”
  - REST APIs
  - Kafka for async events
  - Redis caching
  - Docker deployment