Phase 1: Core Java (Basic to Advanced)

Java Basics

- Syntax, Data Types, Variables, Operators
- Control Structures (if-else, switch, loops)
- Arrays and Strings

Object-Oriented Programming (OOP)

- Classes and Objects
- Inheritance, Polymorphism, Encapsulation, Abstraction
- Interfaces and Abstract Classes

Advanced Java Concepts

- Collections Framework (List, Set, Map)
- Exception Handling
- Java I/O (File handling)
- Generics
- Multithreading and Concurrency
- Java 8 Features (Lambdas, Streams, Optional)

Phase 2: Spring Framework

Basics of Spring Framework

- Introduction to Spring Framework
- Inversion of Control (IoC) and Dependency Injection (DI)
- Setting up a Spring project

Spring Core

- Beans and Bean Factory
- Application Context
- Spring Configuration: XML and Java-based configuration
- Spring Annotations

Spring AOP (Aspect-Oriented Programming)

- Introduction to AOP
- AOP concepts: Aspect, Join Point, Advice, Pointcut
- Spring AOP implementation

Spring Data Access

- JDBC with Spring
- Introduction to Spring Data JPA

Phase 3: Spring Boot

Introduction to Spring Boot

- Benefits of Spring Boot
- Setting up a Spring Boot project
- Spring Boot CLI

Spring Boot Core Concepts

- Auto-configuration
- Spring Boot Starters
- Spring Boot Annotations (@SpringBootApplication, @Configuration, @ComponentScan)

Building RESTful Web Services with Spring Boot

- Creating REST APIs
- Spring Boot Controllers
- Handling HTTP requests and responses
- Error handling in Spring Boot

Data Persistence with Spring Boot

- Integrating Spring Data JPA with Spring Boot
- Repository Pattern
- Creating and managing entities

Phase 4: Hibernate

Introduction to Hibernate

- ORM concepts
- Setting up Hibernate

Core Hibernate Concepts

- Hibernate configuration
- Mapping entities to database tables
- Hibernate Annotations

Hibernate CRUD Operations

- Basic CRUD operations
- HQL (Hibernate Query Language)
- Criteria API

Advanced Hibernate Concepts

- Hibernate caching

- Transactions and concurrency control
- Relationships (One-to-One, One-to-Many, Many-to-Many)

Phase 5: Microservices

Introduction to Microservices

- Microservices Architecture
- Benefits and challenges

Building Microservices with Spring Boot

- Creating microservices
- Communication between microservices (REST, gRPC)
- Service Discovery (Eureka)

Microservices Patterns

- Circuit Breaker (Hystrix, Resilience4j)
- API Gateway (Spring Cloud Gateway)
- Config Server (Spring Cloud Config)

Containerization and Orchestration

- Docker Basics
- Kubernetes Basics

Phase 6: Testing with JUnit

Introduction to JUnit

- JUnit Basics
- Writing test cases

Advanced JUnit Concepts

- Parameterized tests
- Test suites

Spring Boot Testing

- Unit testing Spring Boot applications
- Integration testing with Spring Boot
- Mocking in tests (Mockito)

Additional Concepts to Learn

Version Control

- Git Basics
- GitHub/GitLab usage

Build Tools

- Maven or Gradle

CI/CD

- Jenkins, GitHub Actions

What to Avoid

Outdated Technologies

- Avoid learning deprecated frameworks or libraries (e.g., older versions of Java EE)
- Avoid spending too much time on XML-based Spring configurations as annotation-based configurations are more common now

Niche Technologies

- Avoid overly specialized libraries or tools unless they are specifically required for your projects

Practical Projects

Simple CRUD Application

- Build a basic CRUD application using Spring Boot and Hibernate
- Implement RESTful APIs for the application

E-commerce Application

- Develop a small e-commerce application
- Include user management, product catalog, and order processing

Microservices Application

- Create a set of microservices for a complex application
- Implement inter-service communication, service discovery, and load balancing

Resources

Books

- "Effective Java" by Joshua Bloch
- "Spring in Action" by Craig Walls
- "Spring Boot in Action" by Craig Walls
- "Java Persistence with Hibernate" by Christian Bauer and Gavin King

Documentation

- Java Documentation
- Spring Framework Documentation
- Spring Boot Documentation

---

1. Understand Oracle Database Basics

1.1. Introduction to Databases

- **Concepts**: Relational databases, schema, tables, relationships, and normalization.
- **Tools**: SQL*Plus, Oracle SQL Developer.

1.2. Installing and Configuring Oracle Database

- **Installation**: Learn how to install Oracle Database and Oracle SQL Developer.
- **Configuration**: Basic configuration tasks like setting up a user, schema, and basic connectivity.

2. Learn SQL Fundamentals

2.1. Basic SQL Queries

- **SELECT**: Retrieving data from tables.
- **WHERE**: Filtering results.
- **ORDER BY**: Sorting results.
- **GROUP BY**: Aggregating data.
- **JOINs**: Combining data from multiple tables (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN).

2.2. Advanced SQL

- **Subqueries**: Using nested queries.
- **Set Operations**: UNION, INTERSECT, EXCEPT.
- **Window Functions**: ROW_NUMBER(), RANK(), DENSE_RANK(), etc.
- **Common Table Expressions (CTEs)**: WITH clause.
- **Pivoting and Unpivoting Data**.

3. Learn PL/SQL Basics

3.1. Introduction to PL/SQL

- **What is PL/SQL**: Differences between SQL and PL/SQL.
- **PL/SQL Block Structure**: DECLARE, BEGIN, EXCEPTION, END.

3.2. PL/SQL Variables and Data Types

- **Variables**: Declaration and initialization.
- **Data Types**: Scalar types, composite types (records, tables).

3.3. Control Structures

- **Conditionals**: IF-THEN-ELSE, CASE statements.
- **Loops**: FOR, WHILE, and LOOP statements.

3.4. Cursors

- **Implicit Cursors**: Basic SELECT INTO statement.
- **Explicit Cursors**: Declaring, opening, fetching, and closing cursors.
- **Cursor FOR Loop**: Simplified cursor handling.

3.5. Error Handling

- **Exception Handling**: Using EXCEPTION block to handle runtime errors.

4. Advanced PL/SQL Concepts

4.1. Procedures and Functions

- **Procedures**: Creating and calling procedures.
- **Functions**: Creating and calling functions.
- **Stored Procedures**: Benefits and usage.

4.2. Packages

- **Package Specification and Body**: Defining and using packages.
- **Package Variables**: Using package-level variables.
- **Package Procedures and Functions**: Encapsulation of related procedures and functions.

4.3. Triggers

- **Types of Triggers**: BEFORE, AFTER, INSTEAD OF.
- **Trigger Events**: INSERT, UPDATE, DELETE.
- **Trigger Implementation**: Creating and managing triggers.

4.4. Dynamic SQL

- **Executing Dynamic SQL**: Using EXECUTE IMMEDIATE.
- **Dynamic Queries**: Building and executing queries at runtime.

5. Database Design and Optimization

5.1. Database Design Principles

- **Normalization**: Understanding normal forms.

- **Entity-Relationship Modeling**: Designing database schemas.

5.2. Performance Tuning

- **SQL Optimization**: Analyzing and optimizing SQL queries.
- **Indexes**: Creating and managing indexes.
- **Execution Plans**: Reading and interpreting execution plans.

5.3. Advanced Performance Tuning

- **PL/SQL Optimization**: Optimizing PL/SQL code.
- **Profiling**: Using tools to profile and analyze performance.
- **Database Tuning**: Managing and tuning Oracle database performance.

6. Backup and Recovery

6.1. Backup Strategies

- **Backup Types**: Full, incremental, and cumulative backups.
- **Tools**: RMAN (Recovery Manager).

6.2. Recovery Strategies

- **Recovery Techniques**: Point-in-time recovery, flashback technology.

7. Security

7.1. User and Role Management

- **Users and Roles**: Creating and managing users and roles.
- **Privileges**: Granting and revoking privileges.

7.2. Data Security

- **Encryption**: Encrypting data at rest and in transit.
- **Auditing**: Setting up auditing for security compliance.

8. Additional Resources

8.1. Documentation and Tools

- **Oracle Documentation**: Official Oracle documentation and tutorials.
- **Books**: "Oracle PL/SQL Programming" by Steven Feuerstein, "Oracle Database 12c PL/SQL Programming" by Michael McLaughlin.
- **Online Courses**: Platforms like Udemy, Coursera, or Pluralsight.

8.2. Community and Support

- **Forums**: Oracle Community, Stack Overflow.
- **User Groups**: Oracle User Groups and local meetups.