

FULL STACK JAVA

Introduction to Java Programming

1. Introduction to Java

- Overview of Java language and its features.
- Setting up Java Development Environment (JDK, IDE - IntelliJ IDEA/Eclipse/NetBeans).

2. Java Basics

- Variables, data types, and operators.
- Control flow: if-else, loops, switch.
- Methods, classes, and objects.

3. Object-Oriented Programming (OOP) in Java

- Classes and Objects in detail.
- Inheritance, polymorphism, encapsulation, and abstraction.
- Interfaces and abstract classes.

4. Exception Handling

- Handling exceptions in Java.
- Custom exceptions.

5. Collections Framework

- ArrayList, LinkedList, HashMap, HashSet, etc.
- Iterators and Enumeration.

Advanced Java Concepts

1. Generics

- Introduction to generics.
- Generic classes, methods, and wildcards.

2. Concurrency in Java

- Introduction to threads.
- Synchronization and locks.
- Executors and Thread pools.

3. File Handling

- Reading and writing files.
- Working with streams.

FULL STACK JAVA

4. Java Database Connectivity (JDBC)

- Connecting to databases.
- Executing queries and updates.
- Prepared Statements and Transactions.

5. Java Reflection

- Understanding reflection in Java.
- Dynamic instantiation and method invocation.

Front-end Development with Java

1. Introduction to Front-end Development

- HTML, CSS, JavaScript basics.

2. JavaServer Pages (JSP)

- Setting up JSP in Java EE or Spring Boot.
- Creating dynamic web pages using JSP.

3. Servlets

- Introduction to Servlets.
- Handling HTTP requests and responses.
- Servlet lifecycle.

4. Java Persistence API (JPA)

- Object-Relational Mapping (ORM) with JPA.
- Configuring JPA entities and relationships.

5. Spring MVC

- Introduction to Spring MVC.
- Creating controllers, views, and handling requests.

Full-stack Development with Java

1. Introduction to Spring Boot

- Setting up a Spring Boot project.
- Dependency injection and inversion of control.

2. RESTful Web Services

- Designing RESTful APIs.

FULL STACK JAVA

- Implementing REST endpoints with Spring Boot.
3. **Front-end Framework Integration**
 - Integrating front-end frameworks like Angular, React, or Vue.js with Spring Boot.
 4. **Security**
 - Implementing authentication and authorization using Spring Security.
 - Securing REST APIs.
 5. **Deployment**
 - Deploying Java applications to cloud platforms like AWS, Azure, or Heroku.
 - Containerization with Docker.
 - Continuous Integration and Continuous Deployment (CI/CD) pipelines.