

Back End Developer :- Java

Introduction to Java	What is Java? History of Java Setting up the development environment (JDK, IDEs) Writing your first Java program
Basic Java Syntax	Data types (int, double, char, boolean, etc.) Variables and constants Operators (+, -, *, /, %, etc.) Control statements (if, switch, loops)
Object-Oriented Programming (OOP) Basics	Classes and objects Constructors and methods Inheritance and polymorphism Encapsulation and abstraction
Exception Handling	Handling exceptions with try-catch blocks Creating custom exceptions
Working with Strings	String manipulation and methods StringBuilder and StringBuffer
Collections Framework	Lists, sets, and maps ArrayList, LinkedList, HashSet, HashMap, etc.
File I/O	Reading and writing files Using BufferedReader and BufferedWriter
Advanced OOP Concepts	Interfaces and abstract classes Packages and access modifiers Enums and annotations
Concurrency	Introduction to threads Synchronization Thread safety
Exception Handling	Multiple catch blocks finally block Exception chaining and re-throwing
Java 8 Features	Lambda expressions Streams API Functional interfaces Default methods
Database Connectivity	Connecting to databases

(JDBC)	Executing SQL queries Handling result sets
Unit Testing with JUnit	Writing and running test cases Test fixtures and assertions
Advanced Java Topics	Reflection Generics Design patterns (e.g., Singleton, Factory, Observer)
Spring Framework	Introduction to the Spring framework Dependency Injection Aspect-Oriented Programming (AOP) Spring Boot (if applicable)
Hibernate	Object-Relational Mapping (ORM) Mapping Java objects to database tables Hibernate Query Language (HQL)
Web Development with Java (Servlets and JSP)	Creating web applications Servlet lifecycle JSP basics MVC architecture
RESTful Web Services	Building RESTful APIs with JAX-RS
JavaFX (GUI)	Creating desktop applications with JavaFX
Deployment and Build Tools	Packaging and deploying Java applications Introduction to tools like Maven and Gradle
Security	Java Security features Best practices for secure coding