Coincent 3-Year Program in Java
Partnered by Apsis Solution

Year 1: Live Industrial Training – Build Your Foundation

Gain hands-on industry exposure from day one with 2.5 months of live training in a professional environment. Learn the latest tools and technologies through skill-focused sessions, guided by expert mentors from the industry

Java Curriculum

Chapter 1 – Java Basics (Core Java)

a. Basic Syntax

Covers the structure of a Java program including the main() method, class declaration, semicolons, code blocks, and naming conventions. You'll learn how to write, compile, and run your first Java program.

b. Datatypes and Variables

Introduction to primitive types (int, float, boolean, etc.) and reference types (arrays, objects). Covers declaration, initialization, type casting, and variable scope.

c. Conditions

Decision-making using if, else if, else, and switch-case. Understanding logical operators and nested condition structures.

d. Functions (Methods)

Creating and using methods. Includes method declaration, parameter passing, return values, method overloading, and recursion basics.

e. Loops

Looping constructs such as for, while, do-while, and enhanced for-each. Includes loop control with break and continue.

f. Data Structures

Use of arrays, ArrayList, HashSet, and HashMap. Includes operations like insertion, deletion, searching, and iterating over collections.

g. OOPs (Object-Oriented Programming)

Understanding the pillars of OOP: Classes, Objects, Inheritance, Polymorphism, Abstraction, and Encapsulation. Also includes constructor types and access modifiers.

h. Exception Handling

Techniques to manage runtime errors using try, catch, finally, throw, and throws. Discusses checked vs unchecked exceptions and custom exception classes.

i. Packages

Organizing classes and interfaces into packages. Covers built-in (java.util, java.io) and user-defined packages, and the use of access modifiers.

j. File Handling

Reading from and writing to files using File, Scanner, FileReader, BufferedReader, FileWriter, and PrintWriter. Also includes exception handling in file operations.

Chapter 2 – Advanced Java Concepts

a. JVM (Java Virtual Machine)

Understanding how Java code is executed. Includes JDK vs JRE, class loading, bytecode execution, memory areas, and JIT compiler.

b. Threads

Multithreading concepts, thread lifecycle, creation using Thread and Runnable, thread synchronization, inter-thread communication (wait, notify).

c. Garbage Collection

Memory management using Java's automatic garbage collection. Includes finalize(), reference types, and memory leaks.

d. Generics

Writing reusable and type-safe code using generics in classes, methods, and collections. Covers bounded types and wildcard usage.

e. Streams

Processing collections using Java 8 Stream API. Covers methods like filter(), map(), reduce(), collect(), and usage of parallel streams.

f. Memory Management

JVM memory areas (heap, stack, method area), memory leaks, profiling tools, and garbage collection tuning techniques.

g. Collection Framework

Java's unified framework for storing and manipulating data. Includes List, Set, Map, and Queue interfaces with classes like ArrayList, HashSet, HashMap.

h. Serialization

Converting Java objects into a byte stream using Serializable. Covers transient keyword and versioning using serialVersionUID.

i. Networking and Sockets

Enabling communication between machines using Java's Socket, ServerSocket, and URL classes. Create basic TCP/IP client-server applications.

Chapter 3 – Java Frameworks

a. Build Tools

- **Gradle**
  A powerful build automation tool using Groovy or Kotlin DSL. Faster and more flexible than Maven. Task-based execution.

- **Maven**
  A widely-used project management and build tool that uses XML (POM files) to manage dependencies and project configuration.

b. Web Frameworks (Basics)

- **Spring**
  Core concepts like Inversion of Control (IoC), Dependency Injection (DI), Bean lifecycle, and ApplicationContext.

- **Spring Boot**
  Build stand-alone Spring applications with minimal setup. Includes auto-configuration, embedded servers, and starter dependencies.

c. ORM (Object Relational Mapping)

- **JPA (Java Persistence API)**
  Standard interface for ORM mapping between Java objects and relational databases using annotations like @Entity, @Id.

- **Spring Data JPA**
  A Spring-based abstraction over JPA that provides automated query generation and simplifies repository handling.

- **Hibernate**
  Most popular implementation of JPA, supporting advanced features like lazy/eager loading, caching, and custom query language (HQL).

d. JDBC

- **JDBC Template**
  Part of Spring JDBC. Simplifies database operations like insert, update, and query execution without boilerplate code.

- **JDBI3**
  A modern library for working with relational databases using Java. Combines SQL-friendly and annotation-based configuration.

e. Logging

- **Log4J2**
  A fast and flexible logging framework. Allows logging configuration via XML/JSON/YAML. Supports log levels, appenders, and rolling files.

f. Web Frameworks (Advanced)

1. **Spring Core (Advanced)**
   Deep dive into bean scopes, ApplicationContext, lifecycle callbacks, and property injections.

2. **Spring Security**
   Secures Java applications with features like authentication, role-based access, OAuth2, and JWT token integration.

3. **Spring Data**
   Advanced data access patterns, custom query methods, projections, pagination, and auditing.

4. **Spring MVC**
   REST API creation using annotations like @RestController, @GetMapping, @PostMapping, parameter binding, and response formatting.

Year 2: Real-Time Projects – Apply What You've Learned

Transform your knowledge into real-world experience by working on 8 industry-level projects that build your technical and professional skills. Each project enhances your portfolio, strengthening your resume and showcasing your practical abilities. You'll also collaborate in teams, gaining valuable experience in communication, teamwork, and project management—just like in a real work environment.

PROJECTS:-

Calculator Built in Java
This project involves developing a basic calculator using Java Swing or JavaFX for the GUI. It performs standard arithmetic operations like addition, subtraction, multiplication, and division. The application includes buttons, text fields, and event handling for user interaction. It helps students understand core Java concepts like OOP, event listeners, and exception handling. The UI is designed to be intuitive and responsive. The logic is separated from the presentation layer for modularity. It serves as a foundational Java project for beginners. Tools used: **Java SE**, **Eclipse/IntelliJ**, **Swing/JavaFX**.

Build a Dynamic Website using Java Servlets and JDBC
This project creates a dynamic website where users can register, log in, and interact with a database. It uses **Java Servlets** for handling HTTP requests and **JDBC** for database connectivity (e.g., MySQL). The architecture follows the MVC pattern for clean separation of concerns. HTML/CSS/JavaScript handle the frontend, while Java handles backend logic and sessions. It includes features like form validation, CRUD operations, and user authentication. The project demonstrates real-world web application development. Tools used: **Apache Tomcat**, **JDBC**, **Eclipse**, **MySQL**.

Employee Management System Web App on Spring
This web application allows users to manage employee records—adding, editing, deleting, and searching data. Built using the **Spring Boot** framework, it follows the MVC architecture with Thymeleaf or JSP for views. The backend integrates with a database using **Spring Data JPA** and **Hibernate**. RESTful APIs handle data interaction between the client and server. The project includes role-based access control and input validation. It demonstrates enterprise-level development using Spring ecosystem. Tools used: **Spring Boot**, **JPA**, **MySQL**, **Thymeleaf**, **STS/IntelliJ**.

Year 3 – Placement & Internship Phase:

In the 3rd year of Coincent's program, students are guaranteed an internship with partner companies, complete with a formal Internship Offer Letter and a Completion Certificate upon successful completion. This internship is a complimentary part of the 3-Year "Industrial Training + Internship" model, which also includes live classes, expert mentorship, and hands-on project work. This phase bridges academic learning with real-world application, providing students with valuable professional exposure before graduation.

Coincent also offers structured placement preparation to ensure students are job-ready. This includes portfolio building through 8 real-time projects, certifications aligned with Microsoft standards, and dedicated training for interviews. From mock interviews to resume reviews and HR/technical round prep, every element is designed to transition students from classroom learning to career success. By the 4th year, students are equipped not just with knowledge, but with experience, credentials, and confidence to enter the workforce.