

Java Online Training Course

Our Java course offers a comprehensive, hands-on approach to learning one of the most widely-used programming languages. Designed for beginners and advanced learners alike, the course covers core Java concepts, object-oriented programming, data structures, and real-world application development. Taught by experienced instructors, it ensures students gain practical coding skills to excel in web development, mobile applications, and enterprise solutions.

Course Duration: 2 Month

Flexible Timing: Weekdays & Weekend

Minimum Batch Size

100 % Placement Assistance

Key Features:

Instructor-led Live Online Training

Hands-on Learning

Flexible Schedules

Experienced Trainers

100% Job Assistance

Real-Time Projects

Study Materials

Interview Preparation Guidelines

Global Certification Support

Java Online Course Prerequisites:

To start a Java course, familiarity with basic computer operations is helpful. While prior programming knowledge isn't mandatory for beginner courses, understanding foundational concepts like variables, loops, and conditionals is advantageous. A logical mindset, problem-solving skills, and comfort with math will support success in mastering Java's structure and syntax.

Java Course Syllabus

Core Java Syllabus

Module 1: Introduction to Java

- Why use Java? What is Java?
- Different versions of Java and their features
- Key differences between Java and other programming languages

Module 2: Java Programming Environment

- Setting up Java: Install & configure Java paths
- Writing and running a simple Java program
- Java program structure and compilation process
- Understanding syntax errors, runtime errors, and exceptions
- Overview of JDK, JRE, JVM, and JIT
- Key features of Java
- Basic tools: Text editors and consoles

Module 3: Java Programming Basics

- Naming conventions in Java
- Java comments, statements, and blocks
- Java identifiers, keywords, and literals
- Data types in Java (primitive and reference types)
- Variables, default values, and type casting
- Java operators and their usage

Module 4: Control Structures

- Working with decision-making structures (if, if-else, switch-case)
- Looping structures (for, while, do-while)
- Practical coding exercises

Module 5: Data Input and Arrays

- Getting input from the user using Scanner and BufferedReader
- Introduction to arrays in Java
- Declaring and initializing arrays, multi-dimensional arrays
- Accessing and manipulating arrays
- Handling exceptions like ArrayIndexOutOfBoundsException
- Exercises on arrays and their usage in Java

Module 6: Object-Oriented Programming (OOP) Basics

- Concepts: Abstraction, Encapsulation, Inheritance, Polymorphism
- Creating and working with classes and objects
- Variables (instance and static), methods, and constructors
- The role of "this" keyword
- Access modifiers and their use (public, private, protected)

Module 7: Command-Line Arguments

- Using command-line arguments in Java programs
- Converting and passing command-line arguments

Module 8: Integrated Development Environments (IDEs)

- Using editors like Eclipse for Java development
- Project setup, compilation, and execution in IDEs

Module 9: Inner Classes

- Introduction to inner classes and their types
- Accessing outer class members from an inner class

Module 10: Inheritance in Java

- Understanding inheritance and subclassing
- Constructor chaining and the "super" keyword
- Reference casting and inheritance

Module 11: Abstract Classes and Interfaces

- Abstract methods, classes, and interfaces
- Implementing multiple interfaces
- Differences between interfaces and abstract classes

Module 12: Polymorphism

- Method overloading, overriding, and polymorphic behavior
- "Is-A" vs. "Has-A" relationships

Module 13: Packages and Importing

- Creating and using packages
- Setting classpaths and managing namespaces

Module 14: Using Predefined Packages

- Working with core Java classes like String, Object, Math, System
- Wrapper classes and their usage
- String manipulation with StringBuilder and StringBuffer

Module 15: New Java Concepts

- Concepts: Autoboxing, static imports, enums
- Working with JAR files

Module 16: Garbage Collection

- Introduction to Java's garbage collection
- Working with Java's garbage collection process

Module 17: Exception Handling

- Handling errors with try, catch, finally blocks
- Creating custom exceptions and using the throw keyword

Module 18: Multithreading

- Basics of creating and managing threads
- Thread states, priorities, and synchronization
- Avoiding deadlocks

Module 19: Input/Output Streams

- Working with Java I/O streams
- File handling, copying, and reading files

Module 20: Serialization

- Introduction to serialization and deserialization in Java
- Transient keyword and usage

Module 21: Collection Framework

- Working with collections: List, Set, Map, Queue
- Sorting and manipulating data structures
- Generics in Java collections

Module 22: Java Features

- New features like lambda expressions, Streams API, functional interfaces, optional classes, and other enhancements.

Advanced Java Syllabus

Module 1: JDBC (Java Database Connectivity)

- Database connections with JDBC
- Executing SQL queries
- Transaction management and batch updates

Module 2: JSP (Java Server Pages)

- Basics of JSP and its lifecycle
- Using JSP elements like scriptlet, expression, declaration tags
- Handling exceptions and using implicit objects in JSP
- MVC pattern implementation in JSP

Module 3: Servlet

- Basics of servlets and their lifecycle
- Servlet API and HTTP request handling
- Session tracking and servlet collaboration

Module 4: Advanced Collection and Concurrency

- Implementing core collection structures (ArrayList, LinkedList, etc.)
- Concurrent programming and thread pooling

ARIYALAM

Module 5: Advanced Multithreading

- Using Executor service for efficient thread management
- Handling large files and multi-threaded processing

Module 6: JavaMail API

- Sending and receiving emails using JavaMail
- Sending HTML-formatted emails

Module 7: Design Patterns

- Key design patterns: Singleton, DAO, DTO, MVC, Factory
- Practical usage in applications

Module 8: JUnit Testing

- Basics of JUnit and writing test cases
- Using assertions and annotations

Module 9: Maven

- Dependency management with Maven
- Project setup and configuration using Maven

Module 10: Project Guidance

- Real-world project guidance
- Applying Java concepts and best practices in a final project

FAQ:

1. What is Java, and why is it popular?

Java is an object-oriented programming language used globally for building applications. Its platform independence, security, and versatility make it a favorite for enterprise-level and Android applications.

2. Do I need prior programming knowledge for a Java certification course?

No prior knowledge is required for most beginner Java courses, though familiarity with basic programming concepts like variables, loops, and conditionals can be helpful for understanding Java faster.

3. What is the difference between Java and JavaScript?

Java is a compiled language primarily for backend development, while JavaScript is a scripting language for web applications. They serve different purposes and are not directly related.

4. How long does it take to become proficient in Java?

With consistent study, it typically takes 2-3 months to reach a basic level of proficiency in Java, but mastering advanced topics can take 6-12 months or more.

5. What are Java's primary applications?

Java is used in web development, mobile apps (especially Android), enterprise applications, and IoT devices due to its platform independence, security, and scalability.

6. What top companies have Jobs in Java?

Accenture, Cognizant, Capgemini, IBM, Infosys, TCS, Wipro, and More.