

**Introduction to Java Application Development**

**Course Number:** JAV-422
**Duration:** 5 days

**Overview**

This Java Application Development Fundamentals training course introduces Java programming and object-oriented (OO) development principles. Attendees learn newer Java features, including the Java 9+ module structure, common APIs like Java Collections Framework and JDBC/JPA, and the Java release cycle and Long Term Support (LTS) releases. This Java course is current to Java 11-17.

**Prerequisites**

All attendees must have working knowledge of a modern programming language; no Java experience is needed.

**Materials**

All Java training students receive comprehensive courseware.

**Software Needed on Each Student PC**

* A recent version of Windows, macOS, or Linux with at least 8 GB RAM
* JDK 11 or later
* The Java tool the students are likely to use after the class (Eclipse or IntelliJ IDEA are recommended)
* Other free software - please contact us if you have purchased this class

**Objectives**

* Understand Java’s importance, uses, strengths and weaknesses
* Understand the release cycle and Long Term Support (LTS) releases
* Understand Java language basics
* Write, compile, and run Java programs
* Use the Java shell (JShell - Java 9+) for interactive programming
* Understand the Object Model and Object Oriented Programming
* Understand and use classes, inheritance, polymorphism
* Create well-designed classes and use them in your Java programs
* Use composition and delegation to create objects from other objects
* Understand and use packages to organize code
* Understand and use Java 9 modules
* Understand interfaces, their importance, and their uses
* Use interfaces to implement abstraction
* Learn good Java coding style
* Create well-structured Java programs
* Compile and execute programs with the JDK development tools and with an Integrated Development Environment (IDE) of your choice
* Use the core Java libraries (java.lang, java.util)
* Understand and use exceptions for error handling
* Understand the basics of using JDBC and JPA, and use them to access databases from Java
* Use the Java Collections Framework, including the new API introduced in Java 9-11
* Use other new features, such as type inference
* Use the new features of Java 11-17, as well as important advanced features of earlier Java versions
* Understand and use basic I/O streams (Optional)

**Outline**

* Java Overview
	+ Language and Platform Features
	+ A Simple Java Program
	+ The Java Release Cycle
	+ Program Lifecycle
	+ The Java SE Development Kit (JDK)
	+ Hello World: A Simple Application
* Class and Object Basics
	+ The Object Model and Object-Oriented Programming
	+ Classes, References, and Instantiation
	+ Adding Data to a Class Definition
	+ Adding Methods (Behavior)
	+ Exploring Types and Object Instances
	+ Introducing your IDE
	+ Writing and Using a Class Definition with Fields and Methods
* More on Classes and Objects
	+ More about Methods
	+ Encapsulation and Access Control, public and private Access
	+ Constructors and Initialization
	+ static Members of a Class
	+ Accessor Methods
	+ Encapsulation / Access Protection
	+ Writing and Using Constructors
	+ Static Members (Optional)
* Flow of Control
	+ Branching: if, if-else, switch
	+ Iteration: while, do-while, for, break, continue
	+ Data Validation
* More about Classes and Objects
	+ Type-safe Enums
	+ Wrapper Classes
	+ Java 8+ Date/Time Support
	+ Working with References
	+ Formatted Output
	+ Using Enums
	+ Working with Dates and Times
	+ Using the Debugger
* Strings, Arrays, and Dates/Times
	+ String, StringBuffer, StringBuilder
	+ Arrays, Primitive Arrays, Arrays of Reference Types
	+ Varargs
	+ Using Strings and Arrays
* Packages and Modules
	+ Package Overview Using Packages to Organize Code
	+ Creating Packages, Package Statements, and Required Directory Structure
	+ Import Statements
	+ Classpath
	+ Java Modules Overview
	+ Defining Modules, Requires, and Exports
	+ Module Path and Classpath Differences and Coexistence
	+ Using Packages
	+ Using Java Modules
* Composition and Inheritance
	+ Using Composition to Deal With Complexity
	+ Composition/HAS-A, Delegation
	+ Using Inheritance to share a Commonality
	+ IS-A, extends Inheriting Features
	+ Constructor Issues
	+ Overriding Methods, @Override, Using Polymorphism
	+ Abstract Classes
	+ Working with Composition (Optional)
	+ Using Inheritance
	+ Polymorphism
* Interfaces
	+ Defining and Implementing Interfaces
	+ Using Interfaces
	+ Default Methods and Static Methods
	+ Using Interfaces to Remove Implementation Dependencies
	+ Using Default and Static Methods
* Exceptions
	+ Exceptions and the Exception Hierarchy
	+ Throwing Exceptions and Checked Exceptions
	+ Try and Catch
	+ Handling Exceptions
	+ Try and catch
	+ Program Flow with Exceptions
	+ Multicatch, Finally, Try-with-resources
	+ Throwing Exceptions
	+ Checked Exceptions
* Java Collections and Generics
	+ The Collections Framework and its API
	+ Collections and Java Generics
	+ Collection, List, Set, Map
	+ Autoboxing
	+ Collections of Object (non-generic)
	+ Using ArrayList, HashSet, and HashMap
	+ For-each Loop
	+ Processing Items With an Iterator
	+ More About Generics
	+ Using Collections and Generics
* Database Access with JDBC and JPA
	+ JDBC Overview
	+ JDBC Defined Types
	+ Example of JBDC Usage
	+ JPA Architecture and Programming View
	+ Mapping Entity Classes with Annotations
	+ Persistence Unit, EntityManagerFactory, and EntityManager
	+ Working with JPA (Find by primary key and inserts)
	+ Mapping an Entity Class
	+ Using JPA
	+ Insert/Query Demo
* I/O Streams (Optional)
	+ I/O Streams Overview
	+ Readers and Writers
	+ Exception Handling
	+ Byte Streams
	+ Working with Files
	+ NIO, Paths, Files
	+ Reading and Writing Files
	+ Using Byte Streams (Optional)
* Additional Language Features (Optional)
	+ Functional Interfaces and Lambda Expressions
	+ Switch Expressions
	+ Other Java Features
	+ Working with Lambdas
* Conclusion