

The designers of the Java programming language borrowed heavily from the C language for basic constructs and syntax. Most Java textbooks and training materials assume that the student is a proficient C or C++ programmer, which means the materials bypass the basic, low-level Java syntax that actually gets all the work done, and focus only on higher-level aspects of class design and the use of Java APIs. This leaves non-C/C++ programmers to learn the syntax and semantics of the actual Java language entirely on their own, while at the same time trying to pick up the higher-level concepts.

This four day course is designed to provide a solid foundation in Java for programmers without syntax experience in a C-based language. Besides learning the basic structure and syntax of the language, students will also learn object-oriented principles and how they are applied in Java applications. In addition, this course covers more advanced features of the language such as abstract classes, interfaces, generics, packages, and exception handling. Finally, students will learn two foundational API libraries: I/O streams and collections. Additional appendices on threads, J2EE, and Eclipse are provided for further study.

Course Objectives:

- Write stand-alone applications using the Java language.
- Apply object-oriented principles such as encapsulation, inheritance, and polymorphism in the Java language.
- Create well-scoped classes using packages and inner classes.
- Write programs which both handle and create exceptions.
- Read and write data using input and output streams.
- Use the Java 2 Collections Framework to work with groups of objects.

Audience: Experienced computer programmers who are moving to object-oriented programming using Java.

Prerequisites: Professional programming experience in a high level language, such as COBOL and Visual Basic. This course assumes no experience in a C-based language such as C, C++, or C#.

Number of Days: 4 days

- | | |
|--|---|
| <ol style="list-style-type: none"> 1. Course Introduction
Course Objectives
Overview
Suggested References 2. Getting Started with J2SE
What is Java?
How to Get Java
A First Java Program
Compiling and Interpreting Applications
The JDK Directory Structure 3. Datatypes and Variables
Primitive Datatypes | <ol style="list-style-type: none"> Declarations
Variable Names
Numeric Literals
Character Literals
String
String Literals
Arrays
Non-Primitive Datatypes
The Dot Operator 4. Operators and Expressions
Expressions
Assignment Operator |
|--|---|

	Arithmetic Operators		Printing to the Console
	Relational Operators		printf Format Strings
	Logical Operators		StringBuilder and StringBuffer
	Increment and Decrement Operators		Methods and Messages
	Operate-Assign Operators (+=, etc.)		toString
	The Conditional Operator		Parameter Passing
	Operator Precedence		Comparing and Identifying
	Implicit Type Conversions		Objects
	The Cast Operator		Destroying Objects
5.	Control Flow		Using the Primitive-Type
	Statements		Wrapper Classes
	Conditional (if) Statements		Autoboxing
	Adding an else if	10.	Inheritance in Java
	Conditional (switch) Statements		Inheritance
	while and do-while Loops		Inheritance in Java
	for Loops		Casting
	A for Loop Diagram		Method Overriding
	Enhanced for Loop		Polymorphism
	The continue Statement		super
	The break Statement		The Object Class
6.	Methods	11.	Advanced Inheritance and
	Methods		Language Constructs
	Calling Methods		Enumerated Types - Pre-Java 5.0
	Defining Methods		Enumerated Types Today
	Method Parameters		More Enumerated Types
	Scope		Abstract Classes
	So, Why All the static?		Interfaces
7.	Object-Oriented Programming		Using Interfaces
	Introduction to Object-Oriented		Comparable
	Programming		Collections
	Classes and Objects		Generics
	Fields and Methods	12.	Packages
	Encapsulation		Packages
	Access Control		The import Statement
	Inheritance		Static Imports
	Polymorphism		CLASSPATH and Import
	Best Practices		Defining Packages
8.	Objects and Classes		Package Scope
	Defining a Class	13.	Exception Handling
	Creating an Object		Exceptions Overview
	Instance Data and Class Data		Catching Exceptions
	Methods		The finally Block
	Constructors		Exception Methods
	Access Modifiers		Declaring Exceptions
	Encapsulation		Defining and Throwing
9.	Using Java Objects		Exceptions

- Errors and RuntimeExceptions
 - Assertions
 - 14. Input/Output Streams**
 - Overview of Streams
 - Bytes vs. Characters
 - Converting Byte Streams to Character Streams
 - File Object
 - Binary Input and Output
 - PrintWriter Class
 - Reading and Writing Objects
 - Basic and Filtered Streams
 - 15. Core Collection Classes**
 - The Collections Framework
 - The Set Interface
 - Set Implementation Classes
 - The List Interface
 - List Implementation Classes
 - The Queue Interface
 - Queue Implementation Classes
 - The Map Interface
 - Map Implementation Classes
 - 16. Appendix A – Introduction to Threads**
 - Non-Threaded Applications
 - Threaded Applications
 - Creating Threads
 - Thread States
 - Runnable Threads
 - Coordinating Threads
 - Interrupting Threads
 - Runnable Interface
 - ThreadGroups
 - 17. Appendix B – J2EE Overview**
 - Introduction to J2EE
 - J2SE Building Blocks
 - Servlets, JSPs, and Web Applications
 - Web Services
 - Enterprise JavaBeans
 - Additional J2EE APIs
 - J2EE Clients
 - The J2EE Platform
 - 18. Appendix C – Eclipse**
 - Introduction to Eclipse
 - Installing Eclipse
 - Running Eclipse for the First Time
 - Editors, Views, and Perspectives
- Setting up a Project
 - Creating a New Java Application
 - Running a Java Application
 - Debugging a Java Application
 - Shortcut Key Sequences
 - More Shortcut Key Sequences
 - Setting the Classpath
 - Importing Existing Java Code into Eclipse