

This hands-on course provides a comprehensive introduction to programming, and builds a solid foundation of programming skills that can be used to master additional programming languages like C, C++, or Java™. In this course you will write, compile, and debug programs in Java.

### Course Objectives:

- Explain what computer programs are and what computer programming is about.
- Write and compile simple computer programs.
- Describe basic computer language data types.
- Interact with computer programs using your terminal screen and keyboard.
- Evaluate expressions used in computer programs.
- Design the sequential execution, and flow of decision making, in a program.
- Write programs that use loops to perform repetitive tasks.
- Design and write procedural programs that use methods.
- Use basic debugging techniques to solve programming problems and increase program quality.
- Use arrays and classes for managing program data.
- Write programs that use files to store and retrieve data.

**Audience:** Web developers, system administrators, technical managers, and anyone wanting to develop programming skills.

**Prerequisites:** No programming experience is required.

**Number of Days:** 4 days

<b>1 Course Introduction</b> Course Objectives Course Overview Using the Workbook Suggested References	Performing Numeric Calculations Formatting Output Decision Making Iteration Commenting Your Source Code Good Programming Style
<b>2 Basic Concepts and Definitions</b> What is a Program? “Hello, world!” The Programming Process Program Files and Program Execution System Programs vs. Application Programs Input - Process - Output Programming Languages Compiler Errors vs. Runtime Errors Development Environments	<b>4 Data Types, Constants, and Variables</b> A Program’s Purpose is to Process Data Computer Memory Data Can Be of Different Types Named Data: Variables Literal Data Assignment Printing Variables
<b>3 Writing Simple Programs</b> Reading Input	<b>5 Screen Output and Keyboard Input</b> Writing to the Screen Characters That Have Special Meaning Some Simple Formatting Reading from the Keyboard

	Prompting and Validating		Multidimensional Arrays
<b>6</b>	<b>Expressions</b>		Array Initialization
	Expressions: Where the Work Gets Done	<b>12</b>	<b>Data Collections – Classes</b>
	Expression Evaluation: The Result		What is a Class?
	Arithmetic Expressions		Object vs. Class
	Relational Expressions		Accessing Object Members
	Where are Relational Expressions Used? And? . . . Or?	<b>13</b>	Using Arrays with Classes
	Precedence and Associativity		<b>Working with Files</b>
<b>7</b>	<b>Decision Making</b>		Terminal I/O and File I/O
	Sequential Execution		Opening Files
	What is Decision Making?		Opening a File for Writing
	Simple Decisions: if		Opening a File for Reading
	Two-Way Decisions: else		Checking for File Open Errors
	Code Blocks		Closing a File
	Nesting Control Statements		Text Files vs. Binary Files
	Multi-Way Decisions: switch		
<b>8</b>	<b>Looping</b>		
	Kinds of Loops		
	Iterative Loops		
	Code Blocks and Loops		
	Nested Loops		
	Conditional Loops		
	Infinite Loops		
<b>9</b>	<b>Methods</b>		
	Programming without Methods		
	Reusable Code in a Method		
	The Starting Point		
	Variable Visibility: Scope		
	Parameters		
	Returning a Value		
	Method Stubs		
	Libraries		
<b>10</b>	<b>Debugging</b>		
	What is Debugging?		
	Commenting Out Code		
	Simple Debugging with Print Statements		
	Making Debugging Print Statements Conditional		
	Programs that Help You Debug Programs		
<b>11</b>	<b>Data Collections – Arrays</b>		
	Scalar Data vs. Data Collections		
	What is an Array?		
	Accessing Array Elements		