Java Programming - Introduction

Course ID #: 1260-001-ZZ-W

Hours: 35

Course Content

Course Description:

Intensive and hands-on, the course emphasizes becoming productive quickly as a Java application developer. This course quickly covers the Java 5.0 language syntax and then moves into the object-oriented features of the language. Students will then use several of the provided API packages, such as I/O streams, collections, Swing GUI programming, threads, and accessing a database with JDBC. The course ends with a chapter on performance tuning with hints and best practices for writing efficient applications. Appendices on sockets, regular expressions and J2EE are also available for further study.

Prerequisites:

None.

Topics:

Lesson 1: Java Introduction

- Conventions in These Notes
- The Java Environment – Overview
- Writing a Java Program
- Obtaining the Java Environment
- Setting Up Your Java Environment
- Creating a Class that Can Run as a Program
  - The main() method
- Useful Stuff Necessary to Go Further
  - System.out.printIn()
- Using the Integrated Development Environment
- Exercise: Running a Simple Java Program
- Using the Java Documentation

Lesson 2: Java Basics

- Basic Java Syntax
  - General Syntax Rules
  - Java Statements
  - Blocks of Code
  - Comments
- Variables
  - Declaring Variables
  - Advanced Declarations
- Data
  - Primitive Data Types
  - Object Data Types
  - Literal Values
- Constants and the final Keyword
- Mathematics in Java
  - Basic Rules
  - Expressions
  - Operator Precedence
  - Multiple Assignments
  - Order of Evaluation
  - Bitwise Operators
  - Compound Operators
  - Expressions that Mix Data Types: Typecasting
- Creating and Using Methods
  - Creating Methods
- Variable Scope
- Exercise: Method Exercise

Lesson 3: Java Objects

- Objects
- Object-oriented Languages
- Object-oriented Programs
- Encapsulation
  - OOP as a Messaging System
- Exercise: Object Definition
  - Creating and Using an Instance of an Object
- References
  - Reference Example
  - Reference Expressions
- Defining a Class
  - Access to Data
- More on Access Terms
- Adding Data Members to a Class
  - Adding Method Members (Functions) to a Class
- Standard Practices for Fields and Methods
  - Order of Elements within a Class
- Java Beans
- Bean Properties
- Exercise: Payroll01: Creating an Employee Class
- Constructors
- Instantiating Objects Revisited
- Important Note on Constructors
- Exercise: Payroll02: Adding an Employee Constructor
- Method Overloading
- Exercise: Payroll03: Overloading Employee Constructors
- The this Keyword
- Using this to Call Another Constructor
- Exercise: Using the this Reference
- Static Elements
- The main Method
- Exercise: Payroll05: A static field in Employee
- Garbage Collection
- Java Packages
- Compiling and Executing with Packages
- Working with Packages
- Exercise: Payroll06: Creating an employees Package
- Variable Argument Lists (varargs)
  - Keyboard I/O Using the Console Class
  - Keyboard Input Without the Console
- Exercise: Payroll07: Using KeyboardReader in Payroll
- String, StringBuffer, and StringBuilder
- Creating Documentation Comments and Using javadoc
  - Javadoc Comments
- Exercise: Payroll08: Creating and Using javadoc Comments
- Primitives and Wrapper Classes
  - Autoboxing and Unboxing

Lesson 4: Comparisons and Flow Control Structures

- Boolean-valued Expressions
- Comparison Operators
- Comparing Objects
- Conditional Expression Examples
- Complex boolean Expressions
- Simple Branching
- The if Statement
- if Statement Examples
  - Absolute Value
  - Random Selection
- Exercise: Game01: A Guessing Game
- Exercise Payroll-Control01: Modified Payroll
- Two Mutually Exclusive Branches
  - The if … else Statement
  - Nested if … else Statements – Comparing a Number of Mutually Exclusive Options
- Exercise: Game02: A Revised Guessing Game
- Comparing a Number of Mutually Exclusive options – The switch Statement
  - The switch Statement
  - switch Statement Examples
- Exercise: Game03: Multiple Levels
- Comparing Two Objects
  - Testing Strings for Equivalence
- Conditional Expression
  - while and do … while Loops
  - for Loops

  - For-Each Loops
- Exercise: Payroll-Control02: Payroll with a Loop
- Exercise: Game04: Guessing Game with a Loop
- Additional Loop Control: break and continue
  - Breaking Out of a Loop
- Continuing a Loop
- Classpath, Code Libraries, and Jar Files
  - Using CLASSPATH
  - Creating a jar File (a Library)
- Exercise: Creating and Using an External Library
- Compiling to a Different Directory

Lesson 5: Arrays

- Defining and Declaring Arrays
- Instantiating Arrays
- Initializing Arrays
- Working with Arrays
- Enhanced for Loops – the For-Each Loop
- Array Variables
- Copying Arrays
- Exercise: Using the args Array
- Exercise: Game-Arrays01: A Guessing Game with Random Messages
- Arrays of Objects
- Exercise: Payroll-Arrays01: An Array of employees
- Multi-Dimensional Arrays
- Multidimensional Arrays in Memory
- Example – Printing a Picture
- Typecasting with Arrays of Primitives

Lesson 6: Inheritance

- Inheritance
- Inheritance Examples
- Payroll with Inheritance
- Derived Class Objects
- Polymorphism
  - Inheritance and References
  - Dynamic Method Invocation
- Creating a Derived Class
- Inheritance Example – A Derived Class
- Inheritance and Access
- Inheritance and Constructors – the super Keyword
- Derived Class Methods that Override Base Class Methods
- Inheritance and Default Base Class Constructors
- The Instantiation Process at Runtime
  - Inheritance and static Elements
- Example – Factoring Person Out of Employee
- Exercise: Payroll-Inheritance01: Adding Types of Employees
- Typecasting with Object References
  - More on Object Typecasts
  - Typecasting, Polymorphism, and Dynamic Method Invocation
- More on Overriding
  - Changing Access Levels on Overridden Methods
  - Redefining Fields
- Object Typecasting Example
- Checking an Object's Type: Using instanceof
- Typecasting with Arrays of Objects
- Exercise: Payroll-Inheritance02: Using the Employee Subclassses
- Other Inheritance-related Keywords
  - abstract
  - final
- Exercise: Payroll-Inheritance03: Making our base classes abstract
- Methods Inherited from Object

Lesson 7: Interfaces

- Interfaces
- Creating an Interface Definition
- Implementing Interfaces
    - Implementing Interfaces – Example
- Reference Variables and Interfaces
    - Calling an Interface Method
- Interfaces and Inheritance
- Exercise: Payroll-Interfaces01
- Some Uses for Interfaces
    - Interfaces and Event-Handling
    - Interfaces and "Pluggable Components"
    - Marker Interfaces
- Annotations
- Annotation Details
- Using Annotations

Lesson 8: Exceptions

- Exceptions
- Handling Exceptions
- Exception Objects
- Attempting Risky Code – try and catch
    - try … catch Blocks and Variable Scope/Initialization
    - Example – An Exception You Must Handle
    - Using Multiple catch Blocks
- Guaranteeing Execution of Code – The finally Block
- Letting an Exception be Thrown to the Method Caller
- Throwing an Exception
- Exercise: Payroll-Exceptions01: Handling NumberFormatException in Payroll
- Exercise: Payroll-Exceptions01, continued
- Exceptions and Inheritance
    - Exception Class Constructors and Methods
- Creating and Using Your Own Exception Classes
- Exercise: Payroll-Exceptions02
- Rethrowing Exceptions
- Initializer Blocks
    - Static Initializer Blocks
- Logging
    - Creating a Logger
    - Logger Hierarchy and Naming
    - Log Handlers
    - Log Formatters
- Log Properties
- Assertions

Lesson 9: Collections

- Collections
- Using the Collection Classes
- Using the Iterator Interface
- Creating Collectible Classes
  - hashCode and equals
  - Comparable and Comparator
- Generics
- Bounded Types
- Extending Generic Classes and Implementing Generic Interfaces
- Generic Methods
- Variations on Generics – Wildcards
- Exercise: Payroll using Generics

Lesson 10: Inner Classes

- Inner Classes, aka Nested Classes
- Inner Class Syntax
- Instantiating an Inner Class Instance from within the Enclosing Class
- Inner Classes Referenced from Outside the Enclosing Class
- Referencing the Outer Class Instance from the Inner Class Code
  - static Inner Classes
- Better Practices for Working with Inner Classes
- Enums
  - Why Another Syntax Element for a Set of Constants
  - Defining an enum Class
  - More Complex Enums