

## **INTRODUCTION TO JAVA PROGRAMMING FOR COBOL PROGRAMMERS**

### **General:**

This course provides experienced COBOL programmers with the ability to utilize the structure and syntax of the object oriented Java programming language for both general business and Internet programming applications. The student is prepared to code, test, and execute Java programs making use of the facilities provided by the language. Because procedural programming uses a different paradigm than object oriented programming, this course concentrates on the application of basic object oriented concepts.

The course can be taught using any development environment including WSAD, RAD and Eclipse. Previous knowledge of a procedural programming language such as COBOL is required.

### **Objectives:**

Upon successful completion of this course, the student will be able to:

- Discuss the basic structure and syntax of the Java Programming Language.
- Create and execute simple stand-alone Java programs and graphical Java APPLETS.
- Define and explain encapsulation, inheritance, and polymorphism.
- Declare and use the native data types.
- Describe and use expressions and operators.
- Code object oriented Java programs using all available control structures.
- Create and use classes and interfaces.
- Share and restrict access to class members.
- Effectively use arrays and references within a Java program.
- Describe and use method overloading.
- Define and use constructors and finalizers.
- Increase software reusability through inheritance.
- Use polymorphism to facilitate Java programming efforts.
- Handle and create exception conditions.
- Use the classes in java.io package to process input and output streams.
- Create graphical applications (optional chapter taught only if time permits).

### **Audience:**

COBOL Programmers, and Programmers experienced in procedural programming.

### **Prerequisites:**

Experience and working knowledge of COBOL or another procedural programming language.

### **Duration:**

Five (5) days including classroom lecture and lab sessions. The course manual includes a description of graphical programming, however this section of the course is optional and will only be taught if time permits.

*INTRODUCTION TO JAVA PROGRAMMING FOR COBOL PROGRAMMERS  
COURSE OUTLINE*

**I. INTRODUCTION**

- A. Features of Java
- B. Java Security
- C. History and Origin of Java
- D. Alternative Internet Technologies
- E. Stand-alone Programs
  - 1. Creation (editor)
  - 2. Compilation (javac)
  - 3. Execution (java)
- F. Applets
  - 1. Creation (editor)
  - 2. Compilation (javac)
  - 3. Execution (browser or appletviewer)

**II. OBJECT ORIENTED CONCEPTS**

- A. Features of Object Oriented Languages
- B. Procedural (COBOL) vs. Object Oriented
- C. Data Abstraction
- D. Encapsulation
- E. Inheritance
- F. Polymorphism
- G. Messages
- H. Effects of OO Approach
- I. Basic OO Design (CRC Cards)

**III. JAVA BASICS**

- A. Java Syntax
- B. Java Program Structure
- C. Simple and Compound Statements
- D. Comments and Readability
- E. Identifier Names
- F. Reserved Keywords
- G. Variable Declarations
- H. Native Data Types
- I. Basic Output

**IV. EXPRESSIONS & OPERATORS**

- A. Operators
  - 1. Precedence
  - 2. Associativity
- B. Expressions
  - 1. Evaluating
  - 2. Side Effects
- C. Operators
  - 1. Assignment
  - 2. Increment/Decrement
  - 3. Type Cast

**V. CONTROL STATEMENTS**

- A. Conditional Statements
  - 1. if-else
  - 2. switch-case (EVALUATE)
- B. Iterative Statements
  - 1. while
  - 2. for
  - 3. break
  - 4. continue
- C. Operators
  - 1. Comparison
  - 2. Logical
- D. Basic Input

**VI. CLASSES AND PACKAGES**

- A. Packages
  - 1. Uses
  - 2. Importing
  - 3. Creating
  - 4. classpath variable
- B. Defining Classes
  - 1. Data Members
  - 2. Methods Members
- C. Instantiating Classes
- D. Accessing Class Members
  - 1. Member Access Control
  - 2. Access Specifiers

## VII. ARRAYS AND STRINGS

- A. References and Objects
- B. Arrays
  - 1. Declaring Arrays
  - 2. Accessing Array Elements
  - 3. Physical layout of Arrays
  - 4. Arrays of Objects
  - 5. Copying Arrays
  - 6. Multi-dimensional Arrays
- C. Strings
  - 1. String class Methods
  - 2. StringBuffer class Methods
  - 3. Passing Arguments to main

## VIII. CLASS METHODS AND CONSTRUCTORS

- A. Class Methods
  - 1. Method Overloading
  - 2. this Reference
  - 3. Instance vs. Class Methods
- B. Constructors and Finalizers
  - 1. Overloaded Constructors
  - 2. Static Initializers
  - 3. Finalizer Methods

## IX. INHERITANCE AND POLYMORPHISM

- A. Inheritance
  - 1. Concepts and Terminology
  - 2. Syntax
  - 3. Member Access
  - 4. Constructors and Finalizers
- B. Polymorphism
  - 1. Concepts and Terminology
  - 2. Syntax
  - 3. Overriding Methods
- C. Interfaces
  - 1. Purpose for Interfaces
  - 2. Creating Interfaces
  - 3. Using Interfaces

## X. ERROR AND EXCEPTION HANDLING

- A. Exception Handling Model
- B. Exception Class Hierarchy
- C. Raising and Exception
- D. Dealing with Exceptions
  - 1. Catching
  - 2. Passing Up Call Stack
- E. Defining Exception Classes

## XI. INPUT/OUTPUT STREAMS

- A. Standard Streams
  - 1. in
  - 2. out
  - 3. err
- B. The java.io Package
  - 1. InputStream class
  - 2. OutputStream class
  - 3. Reader class
  - 4. Writer class
- C. File I/O
- D. Filtered Streams
- E. Buffered Streams
- F. Data Streams
- G. Character Streams

## XII. OPTIONAL TOPICS (TIME PERMITTING)

- A. Applets
- B. Graphical User Interface
  - Components, Containers, and Layout Managers
- C. Event Handling

## XIII. HANDOUT - USING WSAD, RAD, OR ECLIPSE