Course Content

Course Description:

This course provides experienced 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. The course can be taught using any development environment including RAD and Eclipse. Previous knowledge of the C or C++ programming languages is not required but would be helpful.

Upon successful completion of this course, the student will be able to:
• Discuss basic structure and syntax of the Java Programming Language
• Create, compile, and execute Java programs
• Explain benefits of Object Oriented Programming
• Define and explain encapsulation, inheritance, and polymorphism
• Declare and use Java 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
• Use String and StringBuilder class methods
• Describe and use method overloading
• Define and use constructors and finalizers
• Increase software reusability through inheritance
• Use polymorphism to facilitate programming efforts
• Define and use interfaces
• Handle and create exception classes

Target Student:

Applications Programmers and Systems Programmers.

Prerequisites:

An understanding of object oriented programming concepts would be helpful but is not required.

Topics:

Module 1: Introduction
- Features of Java
- History and Origin of Java
- Three Editions of Java
- Alternative Internet Technologies
- Stand-alone Programs
  - Creation (editor)
  - Compilation (javac)
  - Execution (java)
- Additional JDK tools
- Using an IDE (RAD, Eclipse)

Module 2: Object Oriented Concepts
- Object Oriented Languages
- Objects
- Messages
- Java Beans
- Test Classes
- Object Oriented Terminology
- Procedural vs. Object Oriented
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism
- Advantages of OO Approach
- Basic OO Design (CRC Cards)

Module 3: Java Basics
- Java Syntax
- Java Program Structure
- Simple and Compound Statements
- Comments and Readability
- Identifier Names
- Reserved Keywords
- Variable Categories
- Variable Declarations
- Primitive/Native Data Types
- String Introduction
- Basic Output

Module 4: Expressions & Operators
- Operators
  - Precedence and Associativity
- Expressions
  - Evaluating
  - Side Effects
- Operators
  - Assignment
  - Arithmetic
  - Compound
  - Increment/Decrement
  - Type Cast

Module 5: Classes and Packages
- What is a Class - Sample
- Packages
  - Uses
  - Importing
  - Creating
  - Classpath
- Some Java API Packages
- Static Imports
- Defining Classes
  - Data Members
  - Methods Members
  - Member Access Control
  - static and final variables
- Instantiating Classes
- Referencing Class Members
- Sample Class Definition and Use

Module 6: Control Statements
- Conditional Statements
  - if-else
  - ternary if
  - switch-case
- Iterative Statements
  - while and do-while
  - for
  - break and continue
- Operators
  - Comparison
  - Logical
- Program Termination
- Basic Input

Module 7: String and StringBuilder
- References and Objects
- Strings and Their Methods
- StringBuilder and StringBuffer
- Contrast String and StringBuilder

Module 8: Arrays and Strings
- What are Arrays?
- Declaring Arrays
- Accessing Array Elements
- For-Each iterator
- Arrays of Objects
- Copying Arrays
- java.util.Arrays Class
- Multi-dimensional Arrays
- ArrayList and Collection Introduction
- Passing Arguments to main

Module 9: Class Methods and Constructors
- Class Methods
  - Method Overloading
  - this Reference
  - Instance vs. Class Data
- Parameter Passing - Primitives vs Objects
- Constructors
  - Overloaded Constructors
  - Constructor Chaining
  - Copy Constructors
  - Static Initializers
- Finalize

Module 10: Inheritance and Polymorphism
- Inheritance
  - Concepts and Terminology
  - Syntax
  - Method Overriding
  - Constructors and Finalize
- Polymorphism
  - Concepts and Terminology
  - Syntax and Example
  - Implications for equals Method
- Interfaces
  - Purpose for Interfaces
  - Creating Interfaces
  - Implementing and Using Interfaces

Module 11: Error and Exception Handling
- Exception Handling Model
- Exception Class Hierarchy
- Checked vs. Unchecked Exceptions
- Raising and Exception
- Handling Exceptions
  - Catching
  - Passing Up Call Stack
- Defining and Using Exception Classes

Module 12: Input/Output Streams (Time permitting)

- Input/Output Stream Model
- Predefined Standard Streams
- Byte-oriented Streams
    - InputStream Class Hierarchy
    - OutputStream Class Hierarchy
- Basic Input and Output Methods
- File I/O
    - File and FileDescriptor
    - Exceptions
- Filtered Streams
- Buffered Streams
- Data Streams
- Object Streams
    - Object Serialization
- Character Streams
    - Reader class
    - Writer class
- Formatting Numbers and Dates
    - NumberFormat
    - DateFormat
    - SimpleDateFormat
    - Formatter and printf
    - Scanner