Java Programming

Course Description

The course teaches the fundamentals of the Java2 programming language, including the syntax of the Java2 language, basic principles of object-oriented programming, exception handling, creating graphical user interfaces with the AWT (abstract windowing toolkit), the Java 1.1 event model, running threads of execution, processing streaming data, and an introduction to networked communication between client and server. Students will learn how to write standalone applications or applets viewable in web browsers. The course is designed for people who are already programmers with other languages, such as C or C++.

Prerequisites

Skills required prior to the course:
- Command-line usage of the classroom operating system, either MS-DOS or UNIX
- Use of an available text editor (for example, Notepad or VI)
- Use a WWW (World Wide Web) browser (for example, Netscape Navigator or MS Internet Explorer)

Highly recommended prior to course:

- Programming experience with a procedural language, such as C or FORTRAN
- Helpful to have prior to the course:
- Programming experience with an object-oriented programming language, such as C++, Objective C, or Smalltalk.
- Experience with object-oriented analysis and design

Course Objectives:

Upon completion of this course, a student will be able to write standalone applications and applets:
- using the Java programming language
- using object-oriented features, such as ordinary inheritance, as well as abstract classes, and interfaces
- using method and data modifiers to control access
- handling exceptions and creating exception classes
- with graphical user interfaces, using Java's AWT
- using the Java 1.1 delegation event model to handle mouse input, text, window, and menu events

Duration

Four and one-half days

Java Programming Outline

1. Module 1: Getting Started

1.1. Overview of Java concepts
1.2. Compile and run a simple Java application

2. Module 2: Java Program Structure

2.1. Introduction to Object-Oriented Paradigm
    2. 1. 1. Abstraction
    2. 1. 1. Encapsulation
    2. 1. 1. Inheritance
    2. 1. 1. Polymorphism
2.2. Comments
2.3. Primitive data types
2.4. Variables
2.5. Methods
    2. 1. 1. Constructors
    2. 1. 1. Overloading methods
2.2. Operators
2.3. Flow-of-control
2.4. Arrays

3. Module 3: Classes and Inheritance

3.1. Inheritance
    3. 1. 1. Superclasses and subclasses and overriding methods
    3. 1. 1. Abstract classes
    3. 1. 1. Interfaces
3.2. Casting reference data types
3.3. Protection
    3. 1. 1. Visibility
3.2. Static methods and data
3.3. Packages
    3. 1. 1. import statement

4. Module 4: Java Class Libraries

4.1. Introduction to using Java classes
4.2. java.lang classes
    4. 1. 1. Object, String, StringBuffer, Math, Number, System,
4.2. java.math classes
    4. 1. 1. BigDecimal, BigInteger
4.2. java.util classes
    4. 1. 1. Date, GregorianCalendar, Random, Vector, Stack, Hashtable, Enumeration, StringTokenizer

5. Module 5: Exception Handling

5.1. Catching and handling exceptions
5.2. Defining exception classes

6. Module 6: AWT (Abstract Windowing Toolkit)

6.1. Create and design a Graphical User Interface (GUI)
    6. 1. 1. GUI components (scroll bars, buttons, etc.)
    6. 1. 1. Layout management (arrangement of components)
6.2. JDK 1.1 Delegation-Based Event Model

6.3. Dialogs
6.4. Menus
6.5. Rendering Graphics

7. Module 7: Applets

7.1. Running Java applets inside a web page
7.2. Applet methods to override
7.3. HTML and applet parameters
7.4. Displaying images (GIF and JPEG formats) and playing audio clips

8. Module 8: Threads

8.1. Concurrency and threads of control
8.2. Thread states
8. 1. 1. Dangers of sharing data
8. 1. 1. Using synchronization to avoid corruption
8. 1. 1. Avoiding deadlocks

9. Module 9: Streams

9.1. Streams and stream classes
9.2. Chaining stream objects
9. 1. 1. Reading and Writing to a File
9.2. Introduction to Serialization
9.3. Properties, including System Properties

10. Module 10: Networking

10.1. Creating a simple TCP client-server connection
10. 1. 1. Communication via Socket objects

11. Summary

11.1. Review of previous materials
11.2. Brief description of advanced topics (what to learn next):
11. 1. 1. Native Methods
11. 1. 1. JDBC
11. 1. 1. what's new in JDK 1.4