



WIRELESS PROGRAMMING USING J2ME AND MIDP

Course Duration: 4 days.

Course Overview: This four-day course introduces experienced Java programmers to the Java 2 Micro Edition, or J2ME, and develops skills in Java programming for wireless devices via the Mobile Information Device Profile, or MIDP. (The MIDP is the “profile” defined within the J2ME for small mobile devices such as cellphones and PDAs.) The course begins with a top-down tour of the J2ME architecture, focusing on wireless programming via the Connected, Limited Device Configuration, or CLDC, and the MIDP. Students learn the simple Core API of the CLDC – primarily by contrast to the Java 2 Standard Edition Core API – and then move into the individual packages of the MIDP.

MIDP study begins with three chapters on user-interface design, which in MIDP is dramatically different from standard Java. Students learn the high- and low-level UI frameworks, and how to use commands and events. Students then study the MIDP Record Management System for limited persistent storage on the device, and then work on mobile networking.

The final module of the course moves beyond basic knowledge of the API to develop effective coding techniques. Students study multithreading, both using the basic Thread/Runnable model and the CLDC’s Timer and TimerTask classes. The bulk of this module is an in-depth treatment of memory management – which is crucial to effective programming on such tiny devices – including techniques for controlling object creation and reuse, string manipulation, and array-based memory management. A third chapter discusses good MIDP programming practices, and the module concludes with a discussion of MIDP in the larger context of networked applications, and particularly with respect to J2EE components.

This course focuses on the application of concepts through substantial hands-on exercises: instructor-led demonstrations and individually-performed labs. A moderately complex MIDP application – a “MIDlet” – is developed over the course as a case study in all the course topics and skills.

LEARNING OBJECTIVES

- Understand the Java 2 Micro Edition architecture, and the stacking of virtual machine, configuration, and profile to address different types of “micro” devices.
- Understand the limitations of mobile devices and see how they drive the design of the Connected, Limited Device Configuration, and the supporting “K” Virtual Machine.
- Identify the key differences between the JVM and the KVM, including build process and code security implementations.

- Understand the design and mission of the Mobile Information Device Profile, and see how programming for mobile devices is fundamentally different from J2SE programming.
- Code cleanly to the CLDC using the KVM.
- Build a simple, functioning MIDlet.
- Understand the framework for packaging and deploying MIDlets to devices.
- Build user interfaces for mobile devices, including text presentation, input controls, 2D graphics, and multi-screen navigation.
- Handle pointer and keypad input in a MIDlet.
- Implement menus and commands.
- Save and re-load information from one MIDlet run to the next using the MIDP Record Management System.
- Make network connections from the mobile device.
- Retrieve remote information via HTTP, including dynamic information based on user requests.
- Use multiple threads and MIDP timers effectively.
- Write memory-conservative code for various tasks, and understand the implications of various techniques on heap usage.
- Understand the use of J2ME and J2EE components to compose a seamless networked application.

Prerequisites: Java programming experience.

COURSE OUTLINE

Module 1. The Java 2 Micro Edition

The J2ME Architecture

- Micro Devices
- Common Limitations
- The Need for J2ME
- The J2ME Software Layer Stack
- Virtual Machine
- Configurations
- Profiles
- The CLDC and the KVM
- Tools and Development Process
- Code Security

The Connected, Limited Device Configuration

- Classification of CLDC Target Devices
- The Core API
- Limitations of Java Language Support in CLDC
- Differences between CLDC and J2SE Packages
- The java.lang Package
- The java.util Package
- CLDC Collections API
- The java.io Package
- CLDC Streams Model
- Using the KVM

The Mobile Information Device Profile

- MIDP Target Devices
- Relationship of MIDP to CLDC
- MIDP Support
- MIDlets
- MIDlet Lifecycle
- Building, Packaging and Deploying MIDlets
- Application Descriptors
- The Application Manager
- Summary of MIDP Packages
- MIDP in Context

Module 2. MIDP Programming

The High-Level User-Interface API

- Presenting a User Interface in a MIDlet
- Organizing a UI by Screens
- The Displayable Hierarchy
- Forms and Items
- Layout Control, or the Lack Thereof
- The TextField Class
- The DateField Class
- The ChoiceGroup Class
- Other Item Classes
- Alerts
- Tickers

The Low-Level User-Interface API

- The Canvas Class
- 2D Graphics
- Fonts
- Drawing Text
- Repainting

Event Handling

- MIDP Event Architecture
- High-Level Event Handling
- Commands
- Item State Changes
- Low-Level Event Handling
- Keypad Input
- Pointer Input
- The Model/View/Controller Pattern
- MVC in Application Design
- Model Events

The Record Management System

- The Challenge of Wireless Persistence
- Remote Storage via MIDP Networking
- Local Storage via the Record Management System
- Scope of Record Management
- Defining a Record
- Writing Information
- Committing Changes
- Reading Information
- Using Streams for Record I/O
- Implementing Object Persistence

Networking

- The CLDC Streams Model
- The CLDC Networking Package
- The MIDP Implementation
- Supported Protocols
- Creating a Connection
- Making an HTTP Request
- Building a CGI String
- Reading Information
- Controlling Download Size
- Using HTTP POST

Module 3. Effective MIDP Programming

Threads and Timers

- The CLDC Threading Model
- Thread and Runnable
- The Timer Class
- The TimerTask Class
- Scheduling TimerTasks

Memory Management

- Importance of Memory Conservation in MIDP
- Memory Management Techniques
- Controlling Object Creation
- Reusing Objects
- Creation on Demand
- Data Compression
- Using Arrays Effectively
- String Manipulation with StringBuffer

Best Practices

- Exception Handling
- Cleaning Up Resources
- UI Design Considerations
- Portability Considerations
- Cross-Platform Development

Mobile Devices in Distributed Systems

- Server-Side Support for Mobile Devices and Client Applications
- The Java 2 Enterprise Platform (J2EE)
- Multi-Tier Architecture
- Java DataBase Connectivity
- Enterprise JavaBeans
- Servlets and JSPs
- PC-Based Fat and Thin Clients
- Mediation for Wireless Clients
- WAP and WML
- Web Services for MIDP

Learning Resources

System Requirements