

J2SE Platform 5.0 for the Developer: Language Changes (WJO-2762)

The J2SE Platform 5.0 for the Developer: Language Changes course provides students with a detailed look at the critical new features in Java 2 Platform, Standard Edition (J2SE platform) 5.0. The course topics include generics, the enhanced for loop, autoboxing and auto-unboxing, typesafe enumerators (enums), variable arguments (varargs), static imports, metadata (also called annotations), formatted input and output (I/O), concurrency, and new networking features. Downloadable labs are provided to enable students to test their understanding of these new features.

Who Can Benefit

Students who can benefit from this course are Java developers that are already familiar with a previous version of the Java language, who require exposure to the new features of J2SE 5.0.



Prerequisites

To succeed fully in this course, students should:

- Understand the fundamentals of the Java programming language
- Understand the principles of object-oriented programming



Skills Gained

Upon completion of this course, students should be able to:

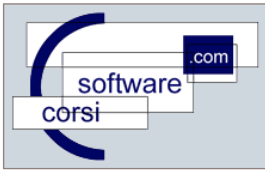
- Understand and identify eight new features in J2SE platform 5.0 including language features and other features.



Related Courses

Before:

- Java Technology for Structured Programmers (SL-265)
- [Java Programming Language \(SL-275\)](#)
- Java Programming Language (WJB-275)



After:

[Java Programming Language Workshop \(SL-285\)](#)
[Java 2 Platform, Enterprise Edition \(J2EE\) Platform Overview for Managers \(WJTB-310\)](#)



Course Content

Module 1 - Overview

- Describe the J2SE platform 5.0 language enhancements
- Describe the new tools available in J2SE platform 5.0
- Describe changes to the core libraries
- Describe enhancements to the Java Virtual Machine (JVM machine)

Module 2 - Generics

- Compare and contrast generic and non-generic collections
- Use the generic version of the Collections API
- Use generics with wildcards
- Write generic methods
- Integrate legacy (non-generic) code with generic code
- Lab: Implement Generics

Module 3 - Enhanced for Loop

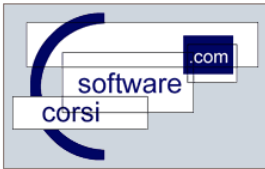
- Use the enhanced for loop with generics
- Use the enhanced for loop with arrays
- Understand nested iterations with the enhanced for loop
- Lab: Use the enhanced for loop to iterate through a collection

Module 4 - Autoboxing and AutoUnboxing

- Describe manual boxing/unboxing and autoboxing/auto-unboxing
- Use autoboxing with generics and the enhanced for loop
- Describe the performance implications of using autoboxing and auto-unboxing
- Lab: Implement autoboxing and auto-unboxing in a sample program

Module 5 - Typesafe Enums

- Describe a common enumerated type pattern before J2SE platform 5.0 and the problems involved
- Create a simple enum
- Create an enum with attributes and constructors
- Create an enum with value-specific methods
- Use enum sets (ranges) and maps



Lab: Write a simple program using typesafe enums

Module 6 - Varargs

Describe the issues with variable-argument methods before J2SE platform 5.0
Use variable argument methods
Create variable argument methods
Lab: Rewrite a sample program to use varargs

Module 7 - Static Import

Describe the purpose of static imports
Use static imports with constants (static attributes), static methods, and enums
Describe the drawbacks of using static imports
Lab: Rewrite a sample program to use static imports

Module 8 - Metadata (Annotations)

Compare annotations before J2SE platform 5.0 with the new annotations
Use built-in annotations
Create your own annotation type
Use your own annotation type in source code
Use your own annotation type at runtime
Describe the Annotation Processing Tool (apt)
Lab: Use annotations to deprecate a method

Module 9 - Formatted Input/Output

Use the PrintWriter format and printf methods in the program
Use the Scanner API
Use the Scanner API with regular expressions
Lab: Implement formatted I/O and the Scanner API

Module 10 - Concurrency Features

Compare the concurrency utilities before J2SE platform 5.0 with the new concurrency utilities
Use an Executor or ExecutorService for thread management
Use Futures and Callables for asynchronous computation
Use locks and conditions as an improvement over the wait and notify methods
Use atomic variables as an improvement over volatile variables
Lab: Implement concurrency features

Module 11 - Networking Enhancements

Describe the fundamental networking changes in J2SE platform 5.0
Create a simple inetd service on Solaris Operating System (Solaris OS)
Create an RMI inetd service on Solaris OS