

Java EE Programming Essentials using IBM Rational Application Developer

Course Summary

Description

This course will introduce the student to the Java programming language, Java class design using UML notation, using RAD 7.0 environment. Through hands-on exercises, the student will become familiar with the Java syntax and the object oriented approach that this language utilizes.

Objectives

By the end of this course, students will be able to:

- Use RAD 7.0 proficiently
- Understand the Java language syntax
- Create, compile, run and debug Java applications
- Manage the Java files
- Translate a class diagram into Java code (and vice-versa)
- Use the Collections framework
- Identify and apply the fundamental design patterns (optional)
- Familiar with new features of Java 5

Topics

- Using RAD 7.0 IDE (can also be adapted to WSAD, RAD or any other IDE)
- Basic Java syntax (data types, operators, control statement)
- Overview of Object Oriented Programming
- Object-oriented programming in Java
- UML notation - Class Diagram
- OOP Using Collections Framework
- Using Exceptions
- Navigating Sun's API documentation
- Generics
- Varargs
- Enhanced Formatting
- Overview of the Fundamental Design Patterns (optional)

Prerequisites

The student must be familiar with basic programming concepts with prior programming experience in at least one language (COBOL, Visual Basic, C, etc.).

Audience

This course is designed for new Java programmers who have not previously programmed in an object-oriented language.

Duration

Five days

Java EE Programming Essentials using IBM Rational Application Developer

Course Outline

I. Java Overview

- A. What is Java?
- B. Java is
- C. Simple and Familiar
- D. Distributed
- E. Dynamic
- F. Robust
- G. Secure
- H. Architectural Neutral
- I. Portable
- J. High Performance
- K. Multithreaded
- L. Interpreted
- M. History of Java
- N. Versions of Java

II. Overview of RAD 7.0

- A. RAD 7.0 Architecture
- B. Getting RAD 7.0
- C. Terminologies
- D. Workbench and Workspace
- E. Workbench
- F. Perspective
- G. View
- H. Editor
- I. Java Development Tooling (JDT)
- J. Java Perspective
- K. File Hierarchy in RAD 7.0
- L. Package Explorer View
- M. Hierarchy View
- N. Outline View
- O. Declaration View
- P. Problem View
- Q. Console View
- R. Java Browsing Perspective
- S. Java Type Hierarchy Perspective
- T. Debug Perspective

III. First Java Application

- A. Java Application
- B. Using the Source Code
- C. Creating HelloWorld in RAD 7.0
- D. Launching RAD 7.0
- E. Creating a new Java Project
- F. Creating a new Class
- G. Saving the Java Class
- H. Running HelloWorld
- I. Comments
- J. Adding Comments in RAD 7.0
- K. Removing Comments in RAD 7.0

IV. Navigating RAD 7.0

- A. Copying a File
- B. Deleting a File
- C. Renaming a File
- D. To Switch Perspectives
- E. To Display Views
- F. Features of Java Editor
- G. To View Source in the Java Editor
- H. Code Assist
- I. Quick Fix
- J. Code Formatting
- K. Adding Required Import Statements
- L. Scrapbook Page
- M. Rearranging the Views
- N. Fast Views
- O. Customizing Java Editor Settings
- P. Bookmarks
- Q. Perspective Preferences
- R. Go to Line Number
- S. Importing and Exporting Files

V. Variables

- A. Declaring Variables
- B. Legal Identifiers
- C. Reserved Words
- D. Integers
- E. Floating Point
- F. Character
- G. Character Constants
- H. Boolean
- I. Declaring Variables
- J. Declarations and Initialization
- K. Declaring Variables
- L. Reference Variables
- M. Assignments & Conversions
- N. Casting
- O. Strings
- P. Literal Types

VI. Operators

- A. Arithmetic Operators
- B. Return type
- C. String Operators
- D. Assignment Operator
- E. Relational Operators

Java EE Programming Essentials using IBM Rational Application Developer

Course Outline (cont'd)

- F. Increment / Decrement Operators
- G. Logical, Bitwise, and Boolean Operators
- H. Logical Not (!)
- I. Short Circuit Operator

VII. Flow of Control

- A. Organizing Statements
- B. Control Statements - if
- C. Control Statements - else
- D. Control Statements - else if
- E. If-else Shortcut
- F. Control Statements - while()
- G. Control Statements do...while()
- H. Control Statements for()
- I. Control Statements - switch
- J. Control Statements - labels
- K. Control Statements - continue
- L. Control Statements - break
- M. Control Statements – return

VIII. Classes and Objects (done)

- A. What is a class ?
- B. Class Declaration Syntax
- C. What is an object ?
- D. Attributes
- E. Attribute Definition Syntax
- F. Methods
- G. Method Definition Syntax
- H. Signature of a Method
- I. Creating an instance of a class
- J. Accessing Attributes and Methods
- K. Scope of Variables
- L. Constructors
- M. Constructor Declaration Syntax
- N. Defining Classes in a Java File
- O. Reference vs. Value
- P. Relational Operators
- Q. == with reference variables
- R. Reference Variables

IX. Arrays

- A. Array Variable Syntax
- B. An Array of Primitive Data Types
- C. An Array of Object Data Types
- D. Dimension of an Array
- E. Arrays - Populating the Array
- F. Array Syntax Shortcut
- G. Referencing the Array

- H. Exceptions in Arrays

- I. The main() method
- J. Arrays of Arrays
- K. Two Dimensional Array

X. Debugging Java Applications Using RAD 7.0

- A. Debug Perspective
- B. The Default Views
- C. Debug Perspective Views
- D. Debug View
- E. Variable View
- F. Display View
- G. Breakpoints View
- H. Java Editor
- I. Expression View
- J. Console View
- K. How to Debug a Java Application
- L. Breakpoints
- M. Hit Count
- N. Launching the Debugger
- O. Evaluating Expressions
- P. Watch
- Q. Display
- R. Inspect

XI. More On Classes

- A. Object-Oriented Programming
- B. Abstraction
- C. Encapsulation
- D. Generating getter/setter Methods
- E. Inheritance
- F. The extends keyword
- G. Using the extends keyword
- H. Polymorphism
- I. Revisiting Reference Variables
- J. Using the this keyword
- K. Using the super keyword

XII. Advanced Features

- A. The package keyword
- B. Package is a Directory Structure
- C. The import keyword
- D. Using final keyword
- E. Using the static keyword
- F. Access Modifiers
- G. The abstract keyword
- H. Using the abstract keyword
- I. The interface keyword

Java 5 Programming Essentials Using IBM RAD

Course Outline (cont'd)

- J. Interface Declaration Syntax
- K. Defining Interfaces in a .java File
- L. Creating an Interface in RAD 7.0
- M. Navigating the Java API
- N. Class String
- O. Class Math
- P. Wrapper Classes
- Q. Enumeration
- R. Enumeration Declaration Syntax
- S. Using Enumeration
- T. More Features of Enumeration
- U. Creating a New Enumeration in RAD 7.0
- V. Static Imports

XIII. Exceptions

- A. What is an Exception?
- B. How Exceptions Work
- C. Error and Exception
- D. Exception Hierarchy
- E. Exception class
- F. Types of Exceptions
- G. Local Exception Handling
- H. Example of Exception Handling
- I. Throwing an Exception
- J. Defining your Own Exception
- K. Handle or Declare the Exception
- L. How it Works Together
- M. Exception Guidelines

XIV. The Collection Framework

- A. Generics
- B. Generics Syntax
- C. For each loop
- D. Collection Interfaces
- E. Priority Queue
- F. Vectors
- G. ArrayList
- H. LinkedList
- I. HashTable
- J. HashSet
- K. TreeSet
- L. HashMap
- M. TreeMap

XV. Streams

- A. I/O Characteristics
- B. What is a Stream?
- C. Predefined Streams
- D. Java I/O Classes
- E. Java.io Classes
- F. Writer Abstract Class
- G. Object Serialization
- H. Writing an Object to a File
- I. Reading an Object from a File
- J. Enhanced Formatting Options
- K. Using the printf() method
- L. Class Formatter