Java SE 7 Programming

Duration: 5 Days

What you will learn

The second of two courses that cover the Java Standard Edition 7 (Java SE 7) Platform, this course covers the core Application Programming Interfaces (API) you will use to design object-oriented applications with Java. Learn to create classes that subclass other classes, extend abstract classes, and program with interfaces. Learn how to properly use exceptions, how to use the Collections framework, and develop applications that manipulate files, directories and file systems. This course also covers writing database programs with JDBC, and how to correctly write multi-threaded applications. Use this course to further develop your skills with the Java language and prepare for the Oracle Certified Professional, Java SE 7 Programmer Exam!
Learn To:

Create Java technology applications with the latest JDK 7 Technology and the NetBeans Integrated Development Environment (IDE)
Enhance object-oriented thinking skills using design patterns and best practices
Identify good practices in the use of the language to create robust Java applications
Manipulate files, directories and file systems
Write database applications using standard SQL queries through JDBC
Create high-performance multi-threaded applications

Audience

Developer
J2EE Developer
Java Developer
Java EE Developer

Prerequisites

Required Prerequisites
Basic understanding of database concepts and SQL syntax

Experience with at least one programming language

Have completed the Java SE 7 Fundamentals course, or experience with the Java language - can create, compile and ex

Understand object-oriented principles

Suggested Prerequisites

Java SE7 Fundamentals

Course Objectives

Perform multiple operations on database tables, including creating, reading, updating and deleting using JDBC technolog
Process strings using a variety of regular expressions
Create high-performing multi-threaded applications that avoid deadlock
Localize Java applications
Create applications that use the Java Collections framework
Implement error-handling techniques using exception handling
Implement input/output (I/O) functionality to read from and write to data and text files and understand advanced I/O strea
Manipulate files, directories and file systems using the JDK7 NIO.2 specification
Apply common design patterns and best practices
Create Java technology applications that leverage the object-oriented features of the Java language, such as encapsulat
Execute a Java technology application from the command line

Course Topics

Java Platform Overview
Introductions
Course Schedule
Java Overview
Java Platforms
OpenJDK
Licensing
Java in Server Environments
The Java Community Process

Java Syntax and Class Review
Simple Java classes
Java fields, constructors and methods
Model objects using Java classes
Package and import statements

Encapsulation and Polymorphism
Encapsulation in Java class design
Model business problems with Java classes
Immutability
Subclassing
Overloading methods
Variable argument methods

Java Class Design
Access modifiers: private, protected and public
Method overriding
Constructor overloading
The instanceof operator
Virtual method invocation
Polymorphism
Casting object references
Overriding Object methods

Advanced Class Design
Abstract classes and type generalization
The static and final modifiers
Field modifier best practices
The Singleton design pattern
Designing abstract classes
Nested classes
Enumerated types

Inheritance with Java Interfaces
Java Interfaces
Types of Inheritance
Object composition and method delegation
Implementing multiple interfaces
The DAO design pattern

Generics and Collections
Generic classes and type parameters
Type inference (diamond)
Collections and generics
List, set and Map
Stack and Deque

String processing
String manipulation with StringBuilder and StringBuffer
Essential String methods
Text parsing in Java
Input processing with Scanner
Text output and formatting
Regular expressions with the Pattern and Matcher classes

Exceptions and Assertions
Exceptions categories
Standard Java Exception classes
Creating your own Exception classes
Using try-catch and the finally clause
Using try-with-resources and the AutoCloseable interface
The multi-catch feature
Best practices using exceptions
Assertions

I/O Fundamentals
I/O using Java
Reading the console input stream
Writing to the console
Using I/O Streams
Chaining I/O Streams
Channel I/O
Reading and writing objects using Serialization

File I/O with NIO 2
The Path interface

The Files class
Directory and File operations
Managing file system attributes
Reading, writing, and creating files
Watching for file system changes

Threading
Operating system task scheduling
Recognizing multithreaded environments
Creating multi-threaded solutions
Sharing data across threads
Synchronization and Deadlock
Immutable objects

Concurrency
Creating Atomic variables
Using Read-Write Locks
Thread-safe collections
Concurrenct synchronizers (Semaphore, Phaser, and others)
Executors and ThreadPools to concurrently schedule tasks
Parallelism and the Fork-Join framework

Database Application with JDBC
Layout of the JDBC API
JDBC divers
Queries and results
PreparedStatement and CallableStatement
Transactions
RowSet 1.1 RowSetProvider and RowSetFactory
The DAO Pattern and JDBC

Localization
Advantages of localization
Defining locale
Read and set locale using the Locale object
Resource bundles
Format messages, dates and numbers