

Java SE 7 Programming

Duration: 5 Days

What you will learn

This Java Programming training covers the core Application Programming Interfaces (API) you'll use to design object-oriented applications with Java. Expert Oracle University instructors will teach you how to write database programs with JDBC.

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.

Create classes that subclass other classes, extend abstract classes and program with interfaces.

Properly use exceptions and the Collections framework.

Develop applications that manipulate files, directories and file systems.

Benefits to You

By enrolling in this course, you'll learn how to boost the productivity, communication and collaboration of your organization. At the same time, you'll develop the knowledge and skills to reduce the cost of application ownership through more efficient development and deployment techniques. Maintain your edge in the job market by staying current with the global standard for developing networked applications.

Earn Your Java Certification

You can use this course to further develop your skills with the Java language. Immersing yourself in this content will help you prepare for the Oracle Certified Professional, Java SE 7 Programmer Exam.

Live Virtual Class Format

A Live Virtual Class (LVC) is exclusively for registered students; unregistered individuals may not view an LVC at any time. Registered students must view the class from the country listed in the registration form. Unauthorized recording, copying, or transmission of LVC content may not be made.

Audience

Developer

Related Training

Required Prerequisites

Experience with at least one programming language

Understand object-oriented principles

Basic understanding of database concepts and SQL syntax

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

Java SE7 Fundamentals

Course Objectives

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 streams

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 encapsulation

inheritance

and polymorphism

Execute a Java technology application from the command line

Perform multiple operations on database tables

including creating

reading

updating and deleting using JDBC technology

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
- Concurrent 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 drivers
- 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