Java SE 25: Programming II Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

Java SE 25: Programming I is designed to expand participants' existing Java programming knowledge by introducing a broad range of advanced development concepts. The course explores secure coding, modular programming, database connectivity, concurrency and multithreading, annotations, advanced generics, and the development and deployment of Java applications on Oracle Cloud.

Using the current Java platform, participants develop the skills required to build more secure, modular, scalable, and high-performing Java applications while working with modern Java APIs and programming practices.

Prerequisites

Participants are expected to have completed **Java SE 25: Programming I** or possess equivalent knowledge and practical Java programming experience.

Who Should Attend

This course is particularly suitable for:

- Data Scientists
- Developers
- Implementation Professionals

What You Will Learn

After completing this course, participants will be able to:

- Develop Java applications using object-oriented concepts including encapsulation, inheritance, and polymorphism.
- Run Java applications from the command line.
- Build applications using the Java Collections Framework.
- Use Lambda Expressions to search and filter collections.
- Apply exception handling techniques to manage application errors.
- Implement Java I/O operations for reading from and writing to data and text files.
- Work with files, directories, and file systems through the JDK NIO.2 specification.
- Perform database table operations using JDBC and JPA technologies.
- Apply concurrency capabilities associated with Lambda Expressions.
- Implement modular programming practices and services within Java applications.
- Develop high-performance multithreaded applications.

Java SE 25: Programming II Training

Duration: **5 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Outline

Java I/O API

- Principles of Java input and output
- Structure and use of the Java I/O API
- Reading and writing binary data
- Fundamental binary data read and write operations
- Reading and writing character data
- Fundamental character data read and write operations
- Connecting streams
- Working with standard input and output
- Using the console
- Understanding serialization
- Serializable object graphs
- Object serialization
- Handling sensitive information during serialization
- Customizing the serialization process
- Serialization and versioning
- Working with file systems
- Constructing file system paths
- Navigating file systems
- Analyzing path properties
- Setting path properties
- Creating paths
- Creating temporary files and directories
- Copying and moving paths
- Deleting paths
- Working with ZIP archives
- Representing a ZIP archive as a file system
- Accessing HTTP resources

Java Concurrency and Multithreading

- Core Java concurrency concepts
- Implementing threads
- Understanding the thread life cycle
- Interrupting threads
- Blocking threads
- Making threads wait for notification
- Common thread properties
- Creating Executor Service objects
- Managing the Executor Service life cycle
- Implementing Executor Service tasks
- Understanding locking problems
- Writing thread-safe code
- Maintaining consistent access to shared data
- Performing non-blocking atomic operations
- Providing exclusive object access with intrinsic locks
- Intrinsic lock automation
- Non-blocking concurrency automation

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Alternative locking mechanisms
- CPU-bound versus I/O-bound concurrent tasks
- Virtual Threads API
- Virtual Thread operations
- Scoped Values
- Working with Scoped Values

Modules and Deployment

- Compiling, packaging, and running non-modular Java applications
- Characteristics of non-modular Java applications
- Understanding modules
- Java Modules
- Java module categories
- Module import declarations
- Defining module dependencies
- Exporting module content
- Module implementation examples
- Opening module content
- Opening an entire module
- Producing and consuming services
- Service implementation examples
- Multi-release module archives
- Compiling and packaging modules
- Running modularized applications
- Migrating legacy Java applications with Automatic Modules
- Creating custom runtime images
- Running runtime images
- Optimizing custom runtime images
- Checking application dependencies

Annotations

- Introduction to annotations
- Designing annotations
- Applying annotations
- Discovering annotations dynamically
- Documenting annotation usage
- Annotations for design validation
- Deprecated annotation
- Suppressing compiler warnings
- Varargs and heap pollution

Java Database Connectivity

- Java Database Connectivity (JDBC)
- JDBC API architecture
- Managing database connections
- Creating and executing basic SQL statements
- Creating and executing prepared SQL statements
- Creating and executing callable SQL statements
- Processing query results

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Managing transactions
- Discovering metadata
- Customizing ResultSet
- Configuring ResultSet types
- Configuring ResultSet concurrency and holdability

Java Security

- Security as a non-functional requirement
- Common security threats
- Denial of Service (DoS) attacks
- Defining security policies
- Changes to the Security API
- Secure file system and I/O operations
- Best practices for protecting application code
- Guarding against erroneous values
- Protecting sensitive data
- Preventing SQL Injection
- Preventing JavaScript Injection
- Preventing XML Injection
- Identifying and documenting security issues

Advanced Generics

- Compiler type erasure for generic information
- Compatibility between generic and raw types
- Generics and type hierarchies
- Wildcard generics
- Upper-bound wildcards
- Lower-bound wildcards
- Best practices for Collections and generics

Java Applications on Oracle Cloud

- Requirements of cloud applications
- Cloud application runtime infrastructure
- Java application servers in cloud environments
- Packaging and deploying cloud applications
- Optimizing deployments with GraalVM
- Fundamentals of the HTTP protocol
- REST service conventions and resources
- Configuring and launching REST service applications with Helidon SE

Additional Advanced Java Topics

- Builder Design Pattern
- Singleton Design Pattern
- Java Regular Expression API
- Regular Expressions: Character Classes
- Regular Expressions: Quantifiers
- Regular Expressions: Boundaries
- File I/O Watch Service

BILGINÇ IT Academy

info@bilginc.com

bilginc.com

- Class-File API
- Foreign Function and Memory API
- Allocating and deallocating memory for native data
- Representing native functions
- Calling native functions

Exams and Assessments

This course covers knowledge and practical skills that can support participants preparing for the separately available **Java SE 25 Developer Professional** exam.