Maslak Mahallesi, Maslak Meydan Sk. No:5. Spring Giz Plaza. Maslak/İstanbul
+90 212 282 7700 - info@bilginc.com

# Java SE 8 New Features

Eğitim Tipi: **Clasroom**

Süre: **2 Day**

## Eğitim Hakkında

This Java SE 8 New Features training delves into the major changes and enhancements in Oracle Java SE 8. Expert Oracle University instructors will help you develop a deeper understanding of the basics; you'll then explore using streams and lambda expressions with collections.Learn To:Understand lambda expressions and streams.Use the new Nashorn JavaScript Engine.Use Mission Control and Flight Recorder.Use new concurrent lambda features.Benefits to YouBy enrolling in this course, you'll develop new knowledge and skills that will help you better leverage the new Java Date and Time API and Nashorn JavaScript engine.

## Önkoşullar

There are no prerequisites for this course.

## Kimler Katılmalı

- Software Engineer
- Software Developers

## Neler Öğreneceksiniz

- Use new concurrent lambda features
- Create lambda expressions using the proper syntax
- Use the new Date/Time API
- Use the new Nashorn JavaScript Engine
- Use Mission Control and Flight Recorder
- Create lambda expressions using the default library interfaces

## Eğitim İçeriği

**Course Introduction**

- Reviewing course objectives
- Discussing course format and LVC
- Getting acquainted with instructor and student
- Discussing course topics planned for coverage
- Overview of changes in 8

**Introducing Lambda Expressions**

- Describing the purpose of an anonymous inner class
- Describing drawbacks to anonymous inner classes
- Describing the components of a lambda expression
- Defining a functional interface
- Creating programs that use lambda expressions

**A Case for Lambda Expressions**

- Discussing the reasons for adding lambda expressions to the Java language
- Reviewing the standard way of extracting data in Java
- Refactoring code to reduce redundancy
- Refactoring code to use inner classes
- Refactoring code to use lambda expressions

- Listing the benefits of lambda expressions

**Filtering Collections with Lambdas**

- Iterating though a collection with forEach
- Iterating through a collection using lambda syntax
- Describing the Stream interface
- Filtering a collection using lambda expressions
- Calling an existing method using a method reference
- Chaining multiple methods together
- Comparing function and imperative programming
- Defining pipelines in terms of lambdas and collections

**Using Built in Lambda Types**

- Listing the built in interfaces included in java.util.function
- Determining true or false with a Predicate
- Processing an object and return nothing with Consumer
- Processing one object and return another with Function
- Generating a new object with Supplier
- Using primitive versions of the base interfaces
- Using binary versions of the base interfaces

**Collection Operations with Lambda**

- Extracting data from an object using map
- Searching for data using search methods
- Describing the types of stream operations
- Describing the Optional class
- Performing calculations using methods
- Describing lazy processing
- Sorting a stream
- Saving results to a collection using the collect method

**Parallel Streams**

- Reviewing the key characteristics of streams
- Contrasting old style loop operations with streams
- Describing how to make a stream pipeline execute in parallel
- Listing the key assumptions needed to use a parallel pipeline
- Defining reduction
- Describing why reduction requires an associative function
- Calculating a value using reduce
- Describing the process for decomposing and then merging work

**Lambda Cookbook**

- Modifying a list using removeIf
- Updating a list using replaceAll
- Updating a map using computeIfAbsent, computerIfPresent, and merge
- Sending the keys and values from a map to a stream
- Reading a file to a stream
- Reading a text file into an ArrayList
- List, walk, and search a directory structure using a stream
- Flattening a stream using flatMap

**Method Enhancements**

- Considering the importance of building good libraries
- Using static methods in Interfaces
- Using default methods
- Understanding default method inheritance rules

**Using the Date/Time API: Working with Local Dates and Times**

- Listing the goals of the Date/Time API (JSR-310)
- Creating and manage date-based events
- Creating and manage time-based events
- Combining date and time into a single object

**Using the Date/Time API: Working with Time Zones**

- Working with dates and times across time-zones and manage changes resulting from daylight savings

**Using the Date/Time API: Working with Date and Time Amounts**

- Defining and create timestamps, periods and durations
- Applying formatting to local and zoned dates and times

**JavaScript on Java with Nashorn: Creating and executing shell scripts**

- Creating and execute shell scripts using JavaScript and Nashorn

**JavaScript on Java with Nashorn: Writing JavaScript Applications**

- Developing JavaScript applications that leverage Java code using Nashorn

**JavaScript on Java with Nashorn: Writing JavaFX Applications Using JavaScript**

- Running JavaScript script from Java applications usingJSR-223
- Prototype JavaFX applications using Nashorn and JavaScript

**Intro to Mission Control**

- Describing JMX and Managed Beans with Mission Control
- Monitoring CPU utilization with Mission Control
- Analyzing JVM characteristics with Mission Control
- Analyzing heap memory with Mission Control

**Intro to Flight Recorder**

- Describing the Java Flight Recorder
- Describing the Java Flight Recorder Architecture
- Starting a Java Flight Recording
- Managing a Java Flight Recording
- Analyzing a Java Flight Recording