

JA-PROG

JAVA SE 8 PROGRAMMING



DURATION	LEVEL	TECHNOLOGY	DELIVERY METHOD	TRAINING CREDITS
5 Days	Professional	Oracle	Classroom Instructor-led Virtual Instructor-led	NA

INTRODUCTION

This five-day course in Java SE 8 Programming thoroughly teaches you the concepts of object-orientation and the tenets of programming using the Java language. You will be taught, *inter-alia* core real-world programming principles and through many practical examples build coding skills to further develop your career in software development. With your newly attained knowledge you will have a solid foundation from which you may develop substantial applications as well as undertake continued skills and training in software development.

AUDIENCE PROFILE

The audience for this course are Java developers, Java application developers, Java mobile developers, Java IOT developers, Java Web application developers and Java technical consultants.

PREREQUISITES

To attend this course, delegates should have:

- Attended Mecer Inter-Ed's Java 8 Fundamentals course.
- Have a good fundamental knowledge of Java 8

COURSE OBJECTIVES

After completing this course, delegates will be able to:

- Thoroughly understand object-oriented concepts such as inheritance, encapsulation and abstraction
- Programme polymorphically
- Develop scalable robust Java technology applications
- Manipulate files, directories and file systems
- Understand functional programming and using Lambda expressions
- Store and manipulate data using collections
- Work with Streams (Stream-map-reduce)
- Create high-performance multi-threaded applications
- Connect to and manipulate databases using standard SQL queries through JDBC
- Be familiar with the material required to take the OCP 1Z0-809 exam

COURSE CONTENT

Module 1: Introduction

This module details all the topics covered throughout the duration of the course.

- Introduction
- Course objectives
- Course schedule

Module 2: Java Environment

In this module you are introduced to the world of Java. You will be shown demonstrations of some Java technologies:

- Java's Technology groups
- Demo:**
- Servlets - Java on the server side

- Internet of things
- The Java community
- Recap the principles of Object orientation.
- Model your objects by learning basic UML

Tasks:

- Answer quiz.
- Design a chess game.

Module 3: Start coding:

In this module you will get yourself ready for programming and will;

- Set up your Java environment
- Write a Java program
- Compile your program

- Becoming familiar with the Java environment.

Tasks:

- Answer quiz.
- Get to know your environment
- Write another application

Module 4: Fundamental Java

This module reviews the core Java fundamentals. The content in the module you should already be familiar with. You will be reminded of;

- Using primitive variables
- Declare Strings
- Manipulating and using Strings

- Working with numbers.
- Understanding variable scope.
- Operators:
- Assignment, relational, incremental, decremental, logical, boolean and bitwise operators

Control Flow:

- The if statement and its companion else.
- Switch statement
- ? operator
- Iteration:
- pre-test and post-test loops.
- Counter and sentinel controlled loops.
- Jumping to the start and out of your loops

Tasks:

- Answer quiz.
- Programming tasks based on all these fundamental concepts.

Module 5: Storing data

This module revises the concept of arrays and the storing and manipulating data. You will learn about;

- Using a single dimension array.
- Using multidimensional arrays.
- Work with command line arguments

Tasks:

- Answer quiz.
- Programming tasks based on arrays.

Module 6: Modelling the world

In this module we discuss the class, how to define them and then construct objects out of them. In this lesson you will learn to;

- Define classes
- Create nested, inner and anonymous classes
- Define class behaviour.
- Pass primitive and complex arguments to a method.
- Return a primitive or complex value from a method.
- Create Immutable classes
- Use enum and Enumeration

Tasks:

- Answer quiz.
- Programming tasks based on class design, immutable classes and enums.

Module 7: Inheritance (is-a)

In this module we continue the concepts of classes and the concept of inheritance. You will learn how to;

- Control access and encapsulate your data.
- Create a subclass
- Understand polymorphism and the principles of overloading and overriding.
- Uses of super and this.

- Cast a superclass reference to the subclass type and visa-versa.
- Design abstract and nonabstract classes
- Create a class hierarchy by extending an abstract class.
- Preventing a class from being subclassed.
- Understand access protection
- Importing packages into your program

Tasks:

- Answer quiz.
- Programming tasks based on inheritance and polymorphism

Module 8: Program to the Interface

This is an important concept in Java and continues the concept of polymorphic programming. In this module you will learn all about;

- Defining an interface
- Implementing interfaces
- Nesting interfaces
- Applying interfaces
- Defining variables in interfaces
- Extending interfaces
- default interface methods
- Multiple inheritance issues
- static methods.

Tasks:

- Answer quiz.
- Programming tasks based on inheritance, polymorphism and interfaces

Module 9: Handling errors – Exception Handling and assertions

In this module we discuss how to handle any abnormal events that may occur during your program's execution. You will learn about;

- Exception types
- Using try, catch and finally
- Autoclosure and try-with-resources
- Displaying a description of an Exception
- Multiple catch clauses
- Nested try statements
- throw and throws keywords
- Java's built-in Exceptions
- Creating your own exceptions
- Using assertions

Tasks:

- Answer quiz.
- Programming tasks based on Exception Handling

Module 10: Java I/O

In this module you will examine Java's I/O system in great detail and learn about Java's basic I/O (input/output) system, including file I/O, serialization; So, you will work with and then understand;

- I/O classes and interfaces

- How to create files and directories
- I/O exceptions
- Byte streams
- Character streams
- The Console class
- Serialization and Serializable

Tasks:

- Answer quiz.
- Programming tasks based on I/O

Module 11: Java NIO.2

In this module you will examine Java's NIO.2 (which is short for New I/O). NIO.2 supports a buffer-oriented, channel-based approach to I/O operations and enhanced support for file handling and file system features. So, you will work with and then understand;

- The Path interface
- The Files class
- The Paths class
- The File attribute interfaces
- The FileSystem, FileSystems, and FileStore classes
- Using the NIO.2 system

Tasks:

- Answer quiz.
- Programming tasks based on NIO.2

Module 12: Generics in Java

In this module you will examine one of the huge changes made to the original Java specification has been the addition of generics. This feature had a profound impact, especially to one of Java's features known as the Collections Framework. We will discuss collections later during the course. Learning about generics before we get there is essential. So, you will work with and then understand;

- The general form of a generic class
- Bounded types
- Wildcard arguments
- Creating a generic method
- Generic constructors and interfaces
- Raw types and legacy code
- Generic class hierarchies
- Overriding methods in a generic class
- Erasure
- Generic restrictions including restrictions on static members

Tasks:

- Answer quiz.
- Programming tasks based on generics

Module 13: Functional Programming with Lambdas

In this module you will learn about Lambda and explore Java 8's built in special interfaces known as functional interfaces. You will get to know and how to use them when

programming Lambdas. You will learn about;

- Lambda expression fundamentals
- Functional interfaces
- Lambda expressions as arguments
- Lambda expressions and exceptions
- Predefined functional Interfaces

Tasks:

- Answer quiz.
- Programming tasks based on Lambda expressions

Module 14: Java's Collection Framework and Wrapper features

In this module we look Java's Collections Framework. A sophisticated hierarchy of interfaces and classes that provide state-of-the-art technology for managing groups of objects. You will understand;

- Autoboxing and type wrappers - using primitive types
- The Collection interfaces
- The List, Set, SortedSet, NavigableSet, Queue and Deque interface
- The Collection classes
- The ArrayList, LinkedList, HashSet, LinkedHashSet, TreeSet, PriorityQueue and ArrayDeque class
- Storing user-defined classes in collections
- Comparators and using a Comparator
- The collection algorithms
- The legacy classes and interfaces
- Properties
- Using store() and load()

Tasks:

- Answer Quiz
- Programming tasks based on collections

Module 15: Streams and filters

You have already learned about Lambda expressions so in this module you will learn the stream API, which is designed with lambda expressions in mind. It provides some of the most significant demonstrations of the power that lambdas bring to Java in a stream's ability to perform very sophisticated operations that search, filter, map or otherwise manipulate data. Using this amazing technology, you will learn;

- Stream Basics
- Stream interfaces
- How to obtain a stream
- How to perform reduction operations
- How to use parallel streams
- How to map your data
- Collect your processed data

- To understand Optional, OptionalDouble, OptionalInt, and OptionalLong

Tasks:

- Answer quiz.
- Programming tasks based on collections, streams and Lambda expressions

Module 16: Threads - multitasking

In this module we will look at Java's built-in support for multithreading and synchronization. It is one of Java's most important innovations major strengths. So, you will learn all about this topic by understanding;

- The Thread Model
- The Thread class
- The Runnable interface
- The main thread
- Creating a Thread - implementing Runnable, extending Thread
- Creating multiple threads
- Using isAlive() and join()
- Thread priorities
- Synchronization
- Interthread communication
- Deadlock
- Suspending, resuming and stopping threads
- Obtaining a thread's state

Tasks:

- Answer quiz.
- Programming tasks based on multitasking

Module 17: Concurrency

Java's original support for multithreading is not ideal for all applications. Java therefore introduced the concept of concurrency, which provides several high-level features, such as synchronizers (such as the semaphore), thread pools, execution managers, locks, several concurrent collections and a streamlined way to use threads to obtain computational results. An important addition was the Fork/Join Framework that facilitates the creation of programs which makes use of multiple processors (such as those found in multicore systems). In this module you will learn about this amazing technology and get to understand;

- The Concurrent API
- Using synchronization objects
- CyclicBarrier
- Using an Executor
- Using Callable and Future
- Locks
- Atomic operations
- Parallel programming via the fork/join framework
- Understanding the impact of the level of parallelism
- Using the RecursiveTask<V>

- Executing a Task asynchronously
- Cancelling a Task
- Determining a Task's completion status
- Restarting a Task

Tasks:

- Answer quiz.
- Programming tasks based on concurrency, forking and joining

Module 18: Data and Time API

In this module we look at a component added in Java 8, which is the new way of handling dates and times in Java. You will understand;

- The Time and Date API
- Formatting Date and Time with java.text
- DateFormat class
- SimpleDateFormat class
- Time and Date fundamentals
- Formatting Date and Time
- Parsing Date and Time strings

Tasks:

- Programming tasks based on Date and Time API

Module 19: JDBC and working with database

In this module we look at a JEE component known as JDBC. JDBC stands for Java database Connectivity. So, you will learn about;

- Relational databases and SQL
- Writing basic SQL statements
- The Interfaces of JDBC
- Connecting to a database
- The JDBC URL
- Obtaining a Database Connection
- Obtaining a Statement and executing a Statement
- Performing CRUD operations
- Getting, reading and scrolling through a ResultSet
- Closing database resources

Tasks:

- Answer quiz.
- Programming tasks based on databases

Module 20: Internationalisation

In this module we look at a component added in Java 8, which is the new way of handling dates and times in Java. You will understand;

- TimeZone
- SimpleTimeZone
- Locale
- ResourceBundle, ListResourceBundle, and PropertyResourceBundle classes

Tasks:

- Programming tasks based on Internationalisation

ADDITIONAL CONTENT

At the end of the course a small project will be given to each delegate that they may attempt once leaving the course to hone their newly acquired skill.

ASSOCIATED CERTIFICATIONS & EXAM

This course will prepare delegates to write the Oracle Certified Professional Java SE 8 Programmer II (1Z0-809) exam.