



---

# 1Z0-810

---

**Upgrade Java SE 7 to Java SE 8 OCP Programmer**  
Exam Summary – Syllabus – Questions



## Table of Contents

<b>Introduction to 1Z0-810 Exam on Upgrade Java SE 7 to Java SE 8 OCP Programmer .....</b>	<b>2</b>
<b>Oracle 1Z0-810 Certification Details:.....</b>	<b>2</b>
<b>Oracle 1Z0-810 Exam Syllabus: .....</b>	<b>2</b>
<b>1Z0-810 Sample Questions: .....</b>	<b>4</b>
<b>Answers to 1Z0-810 Exam Questions: .....</b>	<b>5</b>

# Introduction to 1Z0-810 Exam on Upgrade Java SE 7 to Java SE 8 OCP Programmer

You can use this exam guide to collect all the information about Upgrade Java SE 7 to Java SE 8 OCP Programmer (1Z0-810) certification. The Oracle 1Z0-810 certification is mainly targeted to those candidates who has some experience or exposure of Java and want to flourish their career with Oracle Certified Professional Java SE 8 Programmer (upgrade from Java SE 7) (OCP) credential. The Upgrade Java SE 7 to Java SE 8 OCP Programmer certification exam validates your understanding of the Java technology and sets the stage for your future progression. Your preparation plan for Oracle 1Z0-810 Certification exam should include hands-on practice or on-the-job experience performing the tasks described in following Certification Exam Topics table.

## Oracle 1Z0-810 Certification Details:

Exam Name	Upgrade Java SE 7 to Java SE 8 OCP Programmer
Exam Code	1Z0-810
Exam Product Version	Java SE
Exam Price	USD \$245 (Pricing may vary by country or by localized currency)
Duration	150 Minutes
Number of Questions	60
Passing Score	65%
Format	Multiple Choice
Recommended Training	<a href="#">Java SE 8 New Features</a>
Schedule Exam	<a href="#">Pearson VUE - Oracle</a>
Recommended Practice	<a href="#">1Z0-810 Online Practice Exam</a>

## Oracle 1Z0-810 Exam Syllabus:

TOPICS	
Lambda Expressions	<ul style="list-style-type: none"> <li>- Describe and develop code that uses Java inner classes, including nested class, static class, local class, and anonymous classes</li> <li>- Describe and write functional interfaces</li> <li>- Describe a lambda expression; refactor the code that uses an anonymous inner class to use a lambda expression; describe type inference and target typing</li> </ul>

Using Built-in Lambda Types	<ul style="list-style-type: none"> <li>- Describe the interfaces of the java.util.function package</li> <li>- Develop code that uses the Function interface</li> <li>- Develop code that uses the Consumer interface</li> <li>- Develop code that uses the Supplier interface</li> <li>- Develop code that uses the UnaryOperator interface</li> <li>- Develop code that uses the Predicate interface</li> <li>- Develop code that uses the primitive and binary variations of the base interfaces of the java.util.function package</li> <li>- Develop code that uses a method reference, including refactoring a lambda expression to a method reference</li> </ul>
Java Collections and Streams with Lambdas	<ul style="list-style-type: none"> <li>- Develop code that iterates a collection by using the forEach() method and method chaining</li> <li>- Describe the Stream interface and pipelines</li> <li>- Filter a collection by using lambda expressions</li> <li>- Identify the operations, on stream, that are lazy</li> </ul>
Collection Operations with Lambda	<ul style="list-style-type: none"> <li>- Develop code to extract data from an object by using the map() method</li> <li>- Search for data by using methods such as findFirst(), findAny(), anyMatch(), allMatch(), and noneMatch()</li> <li>- Describe the unique characteristics of the Optional class</li> <li>- Perform calculations by using Java Stream methods, such as count(), max(), min(), average(), and sum()</li> <li>- Sort a collection by using lambda expressions</li> <li>- Develop code that uses the Stream.collect() method and Collectors class methods, such as averagingDouble(), groupingBy(), joining(), and partitioningBy()</li> </ul>
Parallel Streams	<ul style="list-style-type: none"> <li>- Develop code that uses parallel streams</li> <li>- Implement decomposition and reduction in streams</li> </ul>
Lambda Cookbook	<ul style="list-style-type: none"> <li>- Develop code that uses Java SE 8 collection improvements, including Collection.removeIf(), List.replaceAll(), Map.computeIfAbsent(), and Map.computeIfPresent() methods</li> <li>- Develop code that uses Java SE 8 I/O improvements, including Files.find(), Files.walk(), and lines() methods</li> <li>- Use flatMap() methods in the Stream API</li> <li>- Develop code that creates a stream by using the Arrays.stream() and IntStream.range() methods</li> </ul>
Method Enhancements	<ul style="list-style-type: none"> <li>- Add static methods to interfaces</li> <li>- Define and use a default method of an interface and describe the inheritance rules for the default method</li> </ul>
Use Java SE 8 Date/Time API	<ul style="list-style-type: none"> <li>- Create and manage date- and time-based events, including a combination of date and time in a single object, by using LocalDate, LocalTime, LocalDateTime, Instant, Period, and Duration</li> <li>- Work with dates and times across time zones and manage changes resulting from daylight savings, including Format date and times values</li> <li>- Define, create, and manage date- and time-based events</li> </ul>

	using Instant, Period, Duration, and TemporalUnit
--	---

## 1Z0-810 Sample Questions:

**01. What is a common reason for a stream pipeline not to run?**

- a) The source doesn't generate any items.
- b) There are no intermediate operations.
- c) The terminal operation is missing.
- d) None of the above

**02. How do you find out the locale of the running program?**

- a) `Locale.get("default")`
- b) `Locale.get(Locale.DEFAULT)`
- c) `Locale.getDefault()`
- d) None of the above

**03. What technique allows multiple variables from the same class to be shared across all instances of a class?**

- a) Encapsulation
- b) Immutability
- c) Singleton
- d) Static

**04. Under which circumstances does `Files.deleteIfExists()` not throw an exception?**

- a) The file system suddenly becomes unavailable.
- b) The path does not exist.
- c) The path represents a non-empty directory.
- d) The process does not have write access to a path.

**05. Which NIO.2 method is most similar to the legacy `java.io.File.listFiles()` method?**

- a) `Path.listFiles()`
- b) `Files.walk()`
- c) `Files.find()`
- d) `Files.files()`
- e) `Files.list()`
- f) `Files.lines()`

**06. Suppose that you need to work with a collection of elements that need to be sorted in their natural order, and each element has a unique string associated with its value. Which of the following collections classes in the `java.util` package best suit your needs for this scenario?**

- a) Array List
- b) Hash Map
- c) Hash Set
- d) Tree Map
- e) Tree Set
- f) Vector

**07. Which of the following statements are true, assuming a and b are String objects?**

- a) If a.equals(b) is true, a.hashCode() == b.hashCode() is always true.
- b) If a.equals(b) is true, a.hashCode() == b.hashCode() is sometimes but not always true.
- c) If a.equals(b) is false, a.hashCode() == b.hashCode() can never be true.
- d) If a.equals(b) is false, a.hashCode() == b.hashCode() can sometimes be true.

**08. Choose the class that is least likely to be marked Serializable.**

- a) A class that holds data about the amount of rain that has fallen in a given year
- b) A class that manages the memory of running processes in an application
- c) A class that stores information about apples in an orchard
- d) A class that tracks the amount of candy in a gumball machine

**09. What best describes a reduction?**

- a) An intermediate operation where it filters the stream it receives
- b) An intermediate operation where it mathematically divides each element in the stream
- c) A terminal operation where a single value is generated by reading each element in the prior step in a stream pipeline
- d) A terminal operation where one element is returned from the prior step in a stream pipeline without reading all the elements

**10. When localizing an application, which type of data varies in presentation depending on locale?**

- a) Currencies
- b) Dates
- c) Both
- d) Neither

**Answers to 1Z0-810 Exam Questions:**

QUESTION: 01 Answer: c	QUESTION: 02 Answer: c	QUESTION: 03 Answer: d	QUESTION: 04 Answer: b	QUESTION: 05 Answer: e
QUESTION: 06 Answer: d	QUESTION: 07 Answer: a, d	QUESTION: 08 Answer: b	QUESTION: 09 Answer: c	QUESTION: 10 Answer: c

Note: If you find any typo or data entry error in these sample questions, we request you to update us by commenting on this page or write an email on [feedback@oraclestudy.com](mailto:feedback@oraclestudy.com)