1Z0-804

Java SE 7 Programmer II

Exam Summary – Syllabus – Questions

Table of Contents

Introduction to 1Z0-804 Exam on Java SE 7 Programmer II

You can use this document to collect all the information about Java SE 7 Programmer II (1Z0-804) certification. The Oracle 1Z0-804 certification is mainly targeted to those candidates who are from enterprise software development background and want to flourish their career with Oracle Certified Professional Java SE 7 Programmer (OCPJP) credential. The Java SE 7 Programmer II certification exam validates your understanding of the Oracle Java technology and sets the stage for your future progression.

Oracle 1Z0-804 Certification Details:

Exam Name	Java SE 7 Programmer II
Exam Code	1Z0-804
Exam Product Version	Java SE
Exam Price	USD $245 (Pricing may vary by country or by localized currency)
Duration	150 Mins
Number of Questions	66
Passing Score	65%
Validated Against	This exam has been validated against SE 7.
Format	Multiple Choice
Recommended Training	Java SE 7 Programming
Schedule Exam	Pearson VUE - Oracle
Recommended Practice	**1Z0-804 Online Practice Exam**

Oracle 1Z0-804 Exam Syllabus:

Java Class Design	- Use access modifiers: private, protected, and public - Override methods - Overload constructors and methods - Use the instanceof operator and casting - Use virtual method invocation - Override the hashCode, equals, and toString methods from the Object class to improve the functionality of your class. - Use package and import statements
Advanced Class Design	- Identify when and how to apply abstract classes - Construct abstract Java classes and subclasses - Use the static and final keywords - Create top-level and nested classes - Use enumerated types
Object-Oriented Design Principles	- Write code that declares, implements and/or extends interfaces - Choose between interface inheritance and class inheritance - Apply cohesion, low-coupling, IS-A, and HAS-A principles - Apply object composition principles (including has-a relationships) - Design a class using a Singleton design pattern - Write code to implement the Data Access Object (DAO) pattern - Design and create objects using a factory pattern
Generics and Collections	- Create a generic class - Use the diamond for type inference - Analyze the interoperability of collections that use raw types and generic types - Use wrapper classes, autoboxing and unboxing - Create and use List, Set and Deque implementations - Create and use Map implementations - Use java.util.Comparator and java.lang.Comparable - Sort and search arrays and lists
String Processing	- Search, parse and build strings (including Scanner, StringTokenizer, StringBuilder, String and Formatter) - Search, parse, and replace strings by using regular expressions, using expression patterns for matching limited to: . (dot), * (star), + (plus), ?, \d, \D, \s, \S, \w, \W, \b. \B, [], (). - Format strings using the formatting parameters: %b, %c, %d, %f, and %s in format strings.

Exceptions and Assertions	- Use throw and throws statements - Develop code that handles multiple Exception types in a single catch block - Develop code that uses try-with-resources statements (including using classes that implement the AutoCloseable interface) - Create custom exceptions - Test invariants by using assertions
Java I/O Fundamentals	- Read and write data from the console - Use streams to read from and write to files by using classes in the java.io package including BufferedReader, BufferedWriter, File, FileReader, FileWriter, DataInputStream, DataOutputStream, ObjectOutputStream, ObjectInputStream, and PrintWriter
Java File I/O (NIO.2)	- Operate on file and directory paths with the Path class - Check, delete, copy, or move a file or directory with the Files class - Read and change file and directory attributes, focusing on the BasicFileAttributes, DosFileAttributes, and PosixFileAttributes interfaces - Recursively access a directory tree using the DirectoryStream and FileVisitor interfaces - Find a file with the PathMatcher interface - Watch a directory for changes with the WatchService interface
Building Database Applications with JDBC	- Describe the interfaces that make up the core of the JDBC API (including the Driver, Connection, Statement, and ResultSet interfaces and their relationship to provider implementations) - Identify the components required to connect to a database using the DriverManager class (including the jdbc URL) - Submit queries and read results from the database (including creating statements, returning result sets, iterating through the results, and properly closing result sets, statements, and connections) - Use JDBC transactions (including disabling auto-commit mode, committing and rolling back transactions, and setting and rolling back to savepoints) - Construct and use RowSet objects using the RowSetProvider class and the RowSetFactory interface - Create and use PreparedStatement and CallableStatement objects
Threads	- Manage and control thread lifecycle - Synchronize thread access to shared data - Identify code that may not execute correctly in a multi-threaded environment.

Concurrency	- Use collections from the java.util.concurrent package with a focus on the advantages over and differences from the traditional java.util collections. - Use Lock, ReadWriteLock, and ReentrantLock classes in the java.util.concurrent.locks package to support lock-free thread-safe programming on single variables. - Use Executor, ExecutorService, Executors, Callable, and Future to execute tasks using thread pools. - Use the parallel Fork/Join Framework
Localization	- Read and set the locale by using the Locale object - Build a resource bundle for each locale - Call a resource bundle from an application - Format dates, numbers, and currency values for localization with the NumberFormat and DateFormat classes (including number format patterns) - Describe the advantages of localizing an application - Define a locale using language and country codes

1Z0-804 Sample Questions:

01. Sam has designed an application. It segregates tasks that are critical and executed frequently from tasks that are non critical and executed less frequently. He has prioritized these tasks based on their criticality and frequency of execution. After close scrutiny, he finds that the tasks designed to be non critical are rarely getting executed. From what kind of problem is the application suffering?
a) race condition
b) starvation
c) deadlock
d) livelock

02. Given the fragment:
```
public class CustomerApplication {
public static void main (String args[]) {
CustomerDAO custDao= new CustomerDAOMemoryImpl(); // Line 3
// … other methods
}
}
```
Which two valid alternatives to line 3 would decouple this application from a specific implementation of CustomerDAO?
a) CustomerDAO custDao = CustomerDAO();
b) CustomerDAO custDao = (CustomerDAO) new Object ();
c) CustomerDAO custDao = CustomerDAO.getInstance();
d) CustomerDAO custDao = (CustomerDAO) new CustomerDAOmemoryImp1();
e) CustomerDAO custDao = customerDAOFactory.getInstance();

03. Given:
```
public abstract class Account {
abstract void deposit (double amt);
public abstract Boolean withdraw (double amt);
}
```

```
public class CheckingAccount extends Account {
}
```

What two changes, made independently, will enable the code to compile?
a) Change the signature of Account to: public class Account.
b) Change the signature of CheckingAccount to: public abstract CheckingAccount
c) Implement private methods for deposit and withdraw in CheckingAccount.
d) Implement public methods for deposit and withdraw in CheckingAccount.
e) Change Signature of checkingAccount to: CheckingAccount implements Account.
f) Make Account an interface.

04. Given the code fragment:

```
1. Thread t1 = new Thread ();
2. t1.start ()
3. t1.join ( );
4. // . . .
```

Which three are true?
a) On line 3, the current thread stops and waits until the t1 thread finishes.
b) On line 3, the t1 thread stops and waits until the current thread finishes.
c) On line 4, the t1 thread is dead.
d) On line 4, the t1 thread is waiting to run.
e) This code cannot throw a checked exception.
f) This code may throw a checked exception.

05. For which three objects must a vendor provide implementations in its JDBC driver?
a) Time
b) Date
c) Statement
d) ResultSet
e) Connection
f) SQLException
g) DriverManager

06. A valid reason to declare a class as abstract is to:
a) define methods within a parent class, which may not be overridden in a child class
b) define common method signatures in a class, while forcing child classes to contain unique method implementations
c) prevent instance variables from being accessed
d) prevent a class from being extended
e) define a class that prevents variable state from being stored when object Instances are serialized
f) define a class with methods that cannot be concurrently called by multiple threads

07. Which is a factory method from the java.text.NumberFormat class?
a) format (long number)
b) getInstance()
c) getMaxiraumFractionDigits ()
d) getAvailableLocales ()
e) isGroupingUsed()

08. Which two statements are true about Rowset subinterfaces?

a) A JdbcRowSet object provides a JavaBean view of a result set.
b) A cachedRowset provides a connected view of the database.
c) A FilteredRowSet object filter can be modified at any time.
d) A webRowset returns JSON-formatted data.

09. Given:
String s = new String("3"); System.out.print(1 + 2 + s + 4 + 5);
What is the result?
a) 54321
b) 9321
c) 5433
d) 933
e) Compilation fails.

10. You have been asked to create a ResourceBundle file to localize an application. Which code example specifies valid keys menu1 and manu2 with values of File Menu and ViewMenu?
a) <key name ="menu1">File Menu</key>
<key name ="menu1">View Menu</key>
b) <key> menu1</key><File Menu>File Menu </value>
<key> menu1</key><File Menu>View Menu </value>
c) menu1m File menu, menu2, view menu
d) menu1 = File Menu menu2 = View Menu

Answers to 1Z0-804 Exam Questions:

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

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