1Z0-850

Java SE 5 and 6, Certified Associate

Exam Summary – Syllabus – Questions

Table of Contents

Introduction to 1Z0-850 Exam on Java SE 5 and 6, Certified Associate

You can use this document to collect all the information about Java SE 5 and 6 Certified Associate (1Z0-850) certification. The Oracle 1Z0-850 certification is mainly targeted to those candidates who are from enterprise software development background and want to flourish their career with Oracle Certified Associate Java SE 5 and 6 Programmer (OCAJP) credential. The Java SE 5 and 6 Certified Associate certification exam validates your understanding of the Oracle Java technology and sets the stage for your future progression.

Oracle 1Z0-850 Certification Details:

Exam Name	Java SE 5 and 6 Certified Associate
Exam Code	1Z0-850
Exam Product Version	Java SE
Exam Price	USD $245 (Pricing may vary by country or by localized currency)
Duration	115 Mins
Number of Questions	51
Passing Score	68%
Validated Against	This exam has been validated against SE 6.
Format	Multiple Choice
Recommended Training	Fundamentals of the Java Programming Language, Java SE 6 (self-study)
Schedule Exam	Pearson VUE - Oracle
Recommended Practice	**1Z0-850 Online Practice Exam**

Oracle 1Z0-850 Exam Syllabus:

Section 1: Fundamental Object-Oriented Concepts	- Describe, compare, and contrast primitives (integer, floating point, boolean, and character), enumeration types, and objects. - Describe, compare, and contrast concrete classes, abstract classes, and interfaces, and how inheritance applies to them. - Describe, compare, and contrast class compositions, and associations (including multiplicity: (one-to-one, one-to-many, and many-to-many), and association navigation. - Describe information hiding (using private attributes and methods), encapsulation, and exposing object functionality using public methods; and describe the JavaBeans conventions for setter and getter methods. - Describe polymorphism as it applies to classes and interfaces, and describe and apply the "program to an interface" principle.
Section 2: Java Implementation of Object-Oriented Concepts	- Notes: code examples may use the 'new' operator. - Develop code that uses primitives, enumeration types, and object references, and recognize literals of these types. - Develop code that declares concrete classes, abstract classes, and interfaces, code that supports implementation and interface inheritance, code that declares instance attributes and methods, and code that uses the Java access modifiers: private and public. - Develop code that implements simple class associations, code that implements multiplicity using arrays, and recognize code that implements compositions as opposed to simple associations, and code that correctly implements association navigation. - Develop code that uses polymorphism for both classes and interfaces, and recognize code that

	uses the "program to an interface" principle.
Section 3: Algorithm Design and Implementation	- Describe, compare, and contrast these three fundamental types of statements: assignment, conditional, and iteration, and given a description of an algorithm, select the appropriate type of statement to design the algorithm. - Given an algorithm as pseudo-code, determine the correct scope for a variable used in the algorithm, and develop code to declare variables in any of the following scopes: instance variable, method parameter, and local variable. - Given an algorithm as pseudo-code, develop method code that implements the algorithm using conditional statements (if and switch), iteration statements (for, for-each, while, and do-while), assignment statements, and break and continue statements to control the flow within switch and iteration statements. - Given an algorithm with multiple inputs and an output, develop method code that implements the algorithm using method parameters, a return type, and the return statement, and recognize the effects when object references and primitives are passed

	into methods that modify them. - Given an algorithm as pseudo-code, develop code that correctly applies the appropriate operators including assignment operators (limited to: =, +=, -=), arithmetic operators (limited to: +, -, *, /, %, ++, --), relational operators (limited to: <, <=, >, >=, ==, !=), logical operators (limited to: !, &&, ||) to produce a desired result. Also, write code that determines the equality of two objects or two primitives. - Develop code that uses the concatenation operator (+), and the following methods from class String: charAt, indexOf, trim, substring, replace, length, startsWith, and endsWith.
Section 4: Java Development Fundamentals	- Describe the purpose of packages in the Java language, and recognize the proper use of import and package statements. - Demonstrate the proper use of the "javac" command (including the command-line options: -d and -classpath), and demonstrate the proper use of the "java" command (including the command-line options: -classpath, -D and -version). - Describe the purpose and types of classes for the following Java packages: java.awt, javax.swing, java.io, java.net, java.util.
Section 5: Java Platforms and Integration Technologies	- Distinguish the basic characteristics of the three Java platforms: J2SE, J2ME, and J2EE, and given a high-level architectural goal, select the appropriate Java platform or platforms. - Describe at a high level the benefits and basic characteristics of RMI. - Describe at a high level the benefits and basic characteristics of JDBC, SQL, and RDBMS technologies. - Describe at a high level the benefits and basic characteristics of JNDI, messaging, and JMS technologies.

Section 6: Client Technologies	- Describe at a high level the basic characteristics, benefits and drawbacks of creating thin-clients using HTML and JavaScript and the related deployment issues and solutions. - Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating clients using J2ME midlets. - Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating fat-clients using Applets. - Describe at a high level the basic characteristics, benefits, drawbacks, and deployment issues related to creating fat-clients using Swing.
Section 7: Server Technologies	- Describe at a high level the basic characteristics of: EJB, servlets, JSP, JMS, JNDI, SMTP, JAX-RPC, Web Services (including SOAP, UDDI, WSDL, and XML), and JavaMail. - Describe at a high level the basic characteristics of servlet and JSP support for HTML thin-clients. - Describe at a high level the use and basic characteristics of EJB session, entity and message-driven beans. - Describe at a high level the fundamental benefits and drawbacks of using J2EE server-side technologies, and describe and compare the basic characteristics of the web-tier, business-tier, and EIS tier.

1Z0-850 Sample Questions:

01. Which correctly declares a class Test in package com.example?
a) package com.example { class Test { // some code here }
b) package com.example; public class Test { // some code here }
c) public class com.example.Test { // some code here }
d) import com.example; public class Test { // some code here }

02. Which two are features of JNDI? (Choose two.)
a) an interface to store and retrieve named Java objects of any type
b) an interface to search for objects using attributes
c) a defined common set of messaging concepts and programming strategies
d) connectivity to databases and other tabular data sources

03. Which three classes are part of the java.io package? (Choose three.)
a) File

b) Socket
c) URL
d) Reader
e) String
f) BufferedWriter

04. Which is true?
a) You must use JDBC to connect an RDBMS to a Java application.
b) JDBC is designed to provide a bridge between servlets and EJB technology.
c) Classes in the JDBC API include implementations of JDBC drivers.
d) The JDBC API is located within the java.sql and javax.sql packages.

05. Given concrete class B is a subclass of concrete class A, and class A implements interface C, which two are examples of polymorphism? (Choose two.)
a) use a reference variable of type C to refer to an instance of type B
b) use a reference variable of type A to refer to an instance of type C
c) use a reference variable of type C to refer to an instance of type A
d) use a reference variable of type A to refer to an instance of type B
e) use a reference variable of type B to refer to an instance of type A

06. Which two are true? (Choose two.)
a) A single import statement can be used to simplify access to several packages in the Java API
b) A single import statement can be used to simplify access to several classes in the Java API.
c) An import statement is associated with only one class in a source file.
d) A source file can have from zero to many import statements.
e) If a source file has one import statement, it must be preceded by a package statement.

07. You need to create a class Foo that will record the number of times the go() method is invoked on a particular instance of the class. Which solution correctly implements this goal?
a) Declare a local variable invokeCount inside the go() method, and increment the variable within the go() method.
b) Declare a static variable invokeCount for the class Foo, and increment the variable within the go() method.
c) Declare a method parameter invokeCount as the argument to the go() method, and increment the variable within the go() method.
d) Declare an instance variable invokeCount for the class Foo, and increment the variable within the go() method.

08. Which two are true? (Choose two.)
a) Multiplicity indicators are optional, but if they are included they must be shown at both ends of an association.
b) 2..4 is a valid multiplicity indicator.
c) The multiplicity indicators + and 1..* are equivalent.
d) The multiplicity indicators * and 1..* are equivalent.
e) Multiplicity indicators must always be shown at both ends of an association.
f) An optional association is shown using the multiplicity indicator 0..1.

09. Which four are primitive integer types in Java? (Choose four.)
a) float
b) long
c) byte
d) double
e) int
f) short
g) nibble

10. Which is a disadvantage of using J2EE server-side technologies in a web-based application?
a) maintainability
b) complexity
c) support for many different clients
d) scalability

Answers to 1Z0-850 Exam Questions:

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

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