Java

Candidates for this exam are application developers working with Java 6 SE or later, secondary and immediate-post-secondary students of software development, or entry-level software developers.

Candidates should have at least 150 hours of instruction or hands-on experience with Java, be familiar with its features and capabilities, and understand how to write, debug and maintain well-formed, well-documented Java code.

1. Java Fundamentals

1.1 Describe the use of main in a Java application

- Signature of main, how to consume an instance of your own class, command-line arguments

1.2 Perform basic input and output using standard packages

- Print statements, import and use the Scanner class

1.3 Evaluate the scope of a variable

- Declare a variable within a block, class, or method

1.4 Comment and document programs

- Evaluate the syntax of Javadocs, write syntactically correct code comments

2. Data Types, Variables, and Expressions

2.1 Declare and use primitive data type variables

- Data types, including byte, char, int, double, short, long, float, Boolean; identify when precision is lost; initialization; how primitives differ from wrapper object types such as Integer and Boolean

2.2 Construct and evaluate code that manipulates strings

- String class and string literals, comparisons, concatenation, case, and length; String.format methods; string operators; the immutable nature of strings; initialization; null

2.3 Construct and evaluate code that creates, iterates, and manipulates arrays and array lists

- One- and two-dimensional arrays, including initialization, null, size, iterating elements, accessing elements; Collection Framework (ArrayList, Vector, Stack, LinkedList, HashSet) including adding and removing elements, traversing the list

2.4 Construct and evaluate code that performs parsing, casting, and conversion

- Cast between primitive data types, convert primitive types to equivalent object types, parse strings to numbers, convert primitive data types to strings

2.5 Construct and evaluate arithmetic expressions

- Arithmetic operators, assignment, compound assignment operators,

© 2025 Certiport, Inc. Certiport and the Certiport logo are registered trademarks of Certiport Inc. All other trademarks and registered trademarks are the property of their respective holders.

operator precedence

3. Flow Control Implementation

3.1 Construct and evaluate code that uses branching statements

- if, else, else if, switch; single-line vs. block; nesting; logical and relational operators

3.2 Construct and evaluate code that uses loops

- while, for, for each, do while; break and continue; nesting; logical, relational, and unary operators

4. Object-Oriented Programming

4.1 Construct and evaluate class definitions

- Constructors, constructor overloading, one class per .java file, this keyword, basic inheritance and overriding

4.2 Declare, implement, and access data members in classes

- private, public, protected; instance data members; static data members; use static final to create constants; describe encapsulation

4.3 Declare, implement, and access methods

- private, public, protected; method parameters; return type; void; return value; instance methods; static methods; overloading

4.4 Instantiate and use class objects in programs

- Instantiation, initialization, null, access and modify data members, access methods, access and modify static members, import packages and classes

5. Code Compilation and Debugging

5.1 Troubleshoot syntax errors, logic errors, and runtime errors

- Print statements, javac command output, logic errors, console exceptions, stack trace evaluation, using generative AI to find errors in code

5.2 Implement exception handling

- try, catch, finally; Exception class; exception class types; display exception information

© 2025 Certiport, Inc. Certiport and the Certiport logo are registered trademarks of Certiport Inc. All other trademarks and registered trademarks are the property of their respective holders.