hr | avatar
Illuminating Talent

Test Results and Interview Guide

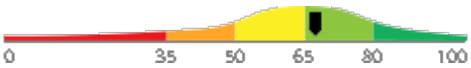
Candidate: **Elizabeth Wantsajob**
Assessment: Java Programming
Completed: June 24, 2026
Prepared for: Sara Maple
Example Company

What's Included

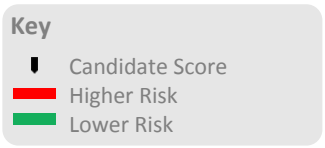
- Overall Score
- Competency Summary Table
- Comparison Matrix
- Detailed Competency Results with Interview Guide

Important Note: The Java Programming assessment measures key factors related to high performance and tenure in this job. Attribute types measured vary by test, but can include cognitive ability, skills, knowledge, personality characteristics, emotional intelligence, and past behavioral history. This report includes a one page summary, followed by detailed results with an embedded interview guide. Note that these results should always be used as a part of a balanced candidate selection process that includes independent evaluation steps, such as interviews and reference checks.

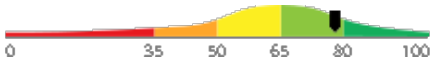
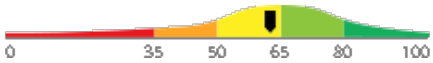
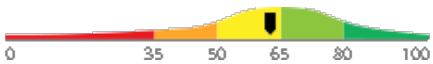
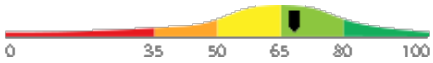
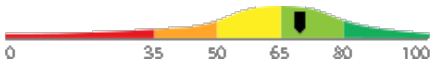
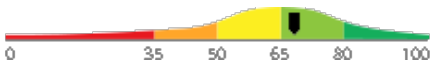
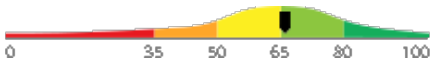
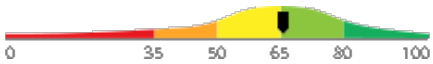
Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com Java Programming June 24, 2026	67	

The candidate demonstrates a solid and competent understanding of Java programming, reflecting proficiency across most core and intermediate-level topics including object-oriented principles, collections, I/O streams, exception handling, and database integration. Minor gaps may exist in specialized areas such as virtual machine monitoring, build utilities, or advanced concurrency, but the candidate is well-suited for entry-level to mid-level development responsibilities.



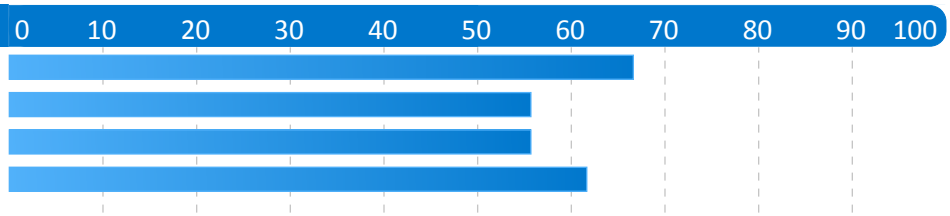
Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Collections Framework and Data Structures	78	
Collections Framework and Data Structures (Coding Tasks)	62	
Object-Oriented Programming (OOP) Concepts (Coding Tasks)	62	
Database Connectivity (JDBC)	68	
Exception Handling	69	
Input/Output (I/O) and Data Serialization	68	
Multithreading and Concurrency	66	
Object-Oriented Programming (OOP) Concepts	66	

Comparison

Percentile scores indicate how the candidate compares to other test-takers within various groups. The candidate scored equal to or better than the fraction of test-takers indicated by the percentile.

Test-Taker Group	Percentile
Global	67th
North America	56th
United States	56th
Example Company	62nd



Artificial Intelligence (AI) Generated Scores

This table includes one or more scores derived from a large language model AI query. AI-derived scores are non-deterministic. That is, they are not precisely repeatable. Therefore, these scores should always be treated as supplementary information and should never be used exclusively or compared to hard cutoff values.

Estimated Value	Score	Confidence	Interpretation
Knowledge, Skills, and Abilities Summary	-	-	Summary Points (AI): • (Generic Text for Sample Report) Strong performer in Drag and Drop Files tasks, indicating comfort with file management and basic computer interactions. • Demonstrates solid numerical accuracy in Recognizing and Confirming Numbers, a valuable asset in detail-oriented roles. • Moderate overall performance in Analytical Thinking and Attention to Detail, with adequate grammar skills but room for improvement. • Struggles with Reading and Analyzing Problems, which may limit effectiveness in roles requiring critical reading and complex problem-solving. • Lowest performance in Navigating Between Screens, suggesting difficulty with multi-screen software workflows that could impact productivity in computer-intensive roles. Narrative (AI): Elizabeth Wantsajob demonstrates a mixed profile of knowledge, skills, and abilities across the assessed competencies. Elizabeth shows a strong aptitude in Drag and Drop Files, performing well on this technical task and suggesting she is comfortable with this type of computer interaction. This is a notable strength that would translate well into roles requiring file management and basic computer navigation tasks. In the area of Analytical Thinking and Attention to Detail, Elizabeth performs at a moderate level. She demonstrates solid ability in Recognizing and Confirming Numbers, which suggests she is careful and accurate when working with numerical data — a valuable skill in detail-oriented work environments. Her Grammar performance is adequate but leaves room for improvement, indicating she may occasionally make written communication errors. Her weakest area within this competency is Reading and Analyzing Problems, where she struggled to consistently interpret and work through written problem scenarios. This may impact her effectiveness in roles that require critical reading, written comprehension, or complex problem-solving. Elizabeth's most significant area for development is Navigating Between Screens, where she scored considerably lower than the other competencies. This suggests she may have difficulty efficiently moving through software interfaces or multi-screen workflows, which could slow productivity in roles that rely heavily on navigating computer applications or data entry systems. Overall, Elizabeth brings some useful technical strengths, particularly in file management and numerical accuracy, but would benefit from targeted development in software navigation and analytical problem-solving to be fully effective in roles that demand these skills. Computed on: April 2, 2026, 11:09:49PM EDT

Detail

Candidate:	Elizabeth Wantsajob, beth.wantsajob@gmail.com
Assessment:	Java Programming
Authorized:	June 24, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
Started:	June 24, 2026, 10:57:24PM EDT
Completed:	June 24, 2026, 10:57:24PM EDT
Overall Score:	67

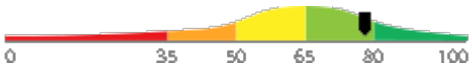
Knowledge and Skills Detail

This section contains a list of job-related knowledge areas and skills that have been evaluated. Low scores in these areas often indicate that additional learning may be required before top performance can be achieved.

Detail | Interview Guide

Collections Framework and Data Structures

Score: 78



Description:

Covers Java's built-in library of data structures, including Lists, Sets, Maps, and Queues, and the selection of the right structure for a given task. Developers use these tools daily to store, retrieve, and manipulate groups of data efficiently in business applications.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates a solid and competent understanding of Java's built-in data structures, including the ability to select and apply appropriate collection types for a variety of business programming tasks. They are expected to work effectively with collections in most day-to-day development scenarios with minimal oversight.

When would you choose a HashMap over a TreeMap in Java, and what are the performance trade-offs between the two?

1	2	3	4	5
Cannot explain the difference or incorrectly describes how either structure works.		Correctly identifies ordering difference but cannot clearly explain hashing or time complexity trade-offs.		Accurately explains hashing vs. tree ordering, O(1) vs. O(log n) access, and gives a practical scenario for each.

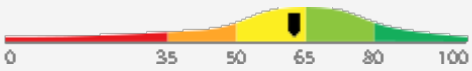
What is the difference between an ArrayList and an array in Java, and when would you use one over the other?

1	2	3	4	5
Cannot clearly distinguish between the two or provides incorrect information about either.		Correctly identifies that ArrayList is resizable but struggles to articulate practical trade-offs.		Clearly explains dynamic sizing, type safety, and performance trade-offs with a practical use case.

Detail

Collections Framework and Data Structures (Coding Tasks)

Score: 62



Description:

Covers the use of pointers to reference and manipulate memory addresses, along with dynamic memory allocation and deallocation using malloc, calloc, realloc, and free. Includes pointer arithmetic, dereferencing, and avoiding common issues like memory leaks and dangling pointers.

Interpretation:

The candidate exhibits average writing skills, which can hinder high performance in some jobs.

The candidate possesses a moderate working knowledge of C programming, demonstrating familiarity with core concepts including data types, control flow, functions, and basic file I/O. They may require some guidance when working with more advanced topics such as dynamic memory allocation, modular design, or debugging complex logic.

Overall AI Score:	65.0
Lines of Code:	15.0
Syntax Errors:	5.0
AI Confidence Level:	50
Match with Ideal Response (AI):	30.0
Structure:	50.0
Syntax:	30.0

Please see below to view the essay submitted.

Interview Guide

Walk me through how you would dynamically allocate memory for an array of 10 integers, use it, and then properly release it. What issues might arise if you don't follow best practices?

1	2	3	4	5
Cannot write correct allocation code; unaware of free() or memory leak risks.		Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.		Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.

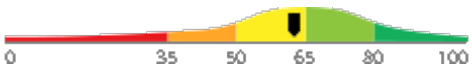
Can you explain what a pointer is in C and describe a situation where you would use one?

1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

Detail

Object-Oriented Programming (OOP) Concepts (Coding Tasks)

Score: 62



Description:

Covers the use of pointers to reference and manipulate memory addresses, along with dynamic memory allocation and deallocation using malloc, calloc, realloc, and free. Includes pointer arithmetic, dereferencing, and avoiding common issues like memory leaks and dangling pointers.

Interpretation:

The candidate exhibits average writing skills, which can hinder high performance in some jobs.

The candidate possesses a moderate working knowledge of C programming, demonstrating familiarity with core concepts including data types, control flow, functions, and basic file I/O. They may require some guidance when working with more advanced topics such as dynamic memory allocation, modular design, or debugging complex logic.

Overall AI Score:	65.0
Lines of Code:	15.0
Syntax Errors:	5.0
AI Confidence Level:	50
Match with Ideal Response (AI):	30.0
Structure:	50.0
Syntax:	30.0

Please see below to view the essay submitted.

Interview Guide

Walk me through how you would dynamically allocate memory for an array of 10 integers, use it, and then properly release it. What issues might arise if you don't follow best practices?

1	2	3	4	5
Cannot write correct allocation code; unaware of free() or memory leak risks.		Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.		Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.

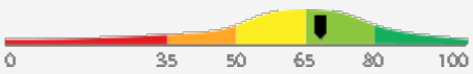
Can you explain what a pointer is in C and describe a situation where you would use one?

1	2	3	4	5
Vague or incorrect definition; cannot describe a practical use case.		Correct basic definition; gives a simple but valid use case with some gaps.		Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

Detail

Database Connectivity (JDBC)

Score: 68



Description:

Covers how Java applications connect to and interact with relational databases using the Java Database Connectivity (JDBC) API, including establishing connections, executing queries, and processing results. Database integration is a core requirement in most business applications for storing and retrieving persistent data.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates a solid and competent understanding of the JDBC API, including establishing connections, executing queries, and processing result sets within Java applications. Minor gaps in knowledge may exist in more advanced scenarios, but the candidate is well-equipped to handle typical database integration requirements in business applications.

Interview Guide

What is the difference between a Statement and a PreparedStatement in JDBC, and why is using a PreparedStatement generally the better choice for business applications?

1	2	3	4	5
Cannot distinguish between the two or incorrectly explains how either works.		Correctly identifies that PreparedStatement uses parameters but cannot clearly explain SQL injection prevention or performance benefits.		Clearly explains parameterized queries, SQL injection prevention, precompilation performance benefits, and gives a practical usage example.

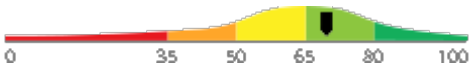
What is JDBC, and can you walk through the basic steps you would take to connect to a database and run a query in Java?

1	2	3	4	5
Cannot define JDBC or describes an incomplete or incorrect sequence of steps.		Correctly identifies JDBC and mentions Connection and Statement but misses steps like loading a driver or closing resources.		Clearly describes loading a driver, obtaining a Connection, creating a Statement, executing a query, and processing ResultSet.

Detail

Exception Handling

Score: 69



Description:

Covers how Java programs detect, report, and recover from errors using try-catch-finally blocks, checked and unchecked exceptions, and custom exception classes. Proper exception handling is essential for writing reliable, production-ready business applications that handle unexpected conditions gracefully.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates a solid and competent understanding of Java exception handling principles, including the use of checked and unchecked exceptions and custom exception classes. They are likely capable of writing reliable business applications that handle unexpected conditions gracefully, with only occasional gaps in more advanced areas.

Interview Guide

What is the difference between a checked and an unchecked exception in Java, and how does that difference affect how you write your code?

1	2	3	4	5
Cannot distinguish between the two types or incorrectly describes how they are handled.		Correctly identifies that checked exceptions must be declared or caught but struggles to explain design implications.		Clearly explains compile-time enforcement of checked exceptions, RuntimeException hierarchy, and design trade-offs with examples.

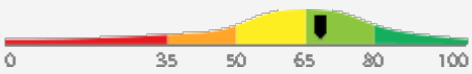
What is an exception in Java, and how would you use a try-catch block to handle one in your code?

1	2	3	4	5
Cannot define exception or demonstrates incorrect syntax for try-catch.		Correctly describes try-catch syntax but cannot explain when or why to use specific exception types.		Clearly explains exceptions, correct try-catch-finally usage, and gives a practical error-handling scenario.

Detail

Input/Output (I/O) and Data Serialization

Score: 68



Description:

Covers reading and writing data using Java's I/O streams, file handling, and the conversion of objects to and from a storable or transferable format through serialization. These capabilities are used regularly in business applications to process files, exchange data between systems, and persist application state.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate demonstrates a solid and competent understanding of Java I/O and data serialization, and is capable of handling most file processing and data exchange tasks effectively. Minor gaps in advanced topics may exist, but they can generally work independently in this knowledge area.

Interview Guide

What is Java serialization, and how would you use it to save an object's state to a file and restore it later?

1	2	3	4	5
Cannot define serialization or incorrectly describes the process or required interfaces.		Correctly identifies the Serializable interface and ObjectOutputStream but cannot explain versioning or exclusions.		Explains Serializable, ObjectOutputStream/InputStream usage, serialVersionUID importance, and transient keyword with a clear example.

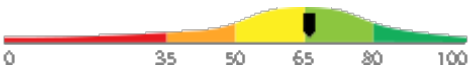
How would you read the contents of a text file in Java, and what classes would you use to do it?

1	2	3	4	5
Cannot name relevant classes or describes an incorrect approach to reading files.		Names a relevant class like BufferedReader or Scanner but cannot explain resource management or error handling.		Correctly describes using BufferedReader or Files utility, explains try-with-resources, and mentions character encoding.

Detail

Multithreading and Concurrency

Score: 66



Description:

Covers the creation and management of threads in Java, including the use of the java.util.concurrent package, synchronization, thread pools, and common concurrency utilities like ExecutorService and locks. These skills are needed to build responsive, high-performance business applications that handle multiple tasks at the same time.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate exhibits a solid and competent understanding of Java multithreading and concurrency, including thread lifecycle management, synchronization strategies, and the use of key concurrency utilities. They are likely capable of building and maintaining responsive, multi-threaded Java applications with moderate complexity.

Interview Guide

How does an ExecutorService help manage threads in a Java application, and why would you use it instead of creating threads manually?

1	2	3	4	5
Cannot explain ExecutorService or incorrectly describes its relationship to threads.		Correctly identifies that ExecutorService manages a thread pool but cannot explain configuration or shutdown procedures.		Clearly explains thread pool management, task submission, resource efficiency, and proper shutdown with a practical example.

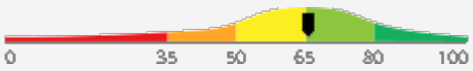
What is a thread in Java, and can you describe a simple way to create and start one in your code?

1	2	3	4	5
Cannot define a thread or confuses threads with processes; unable to describe creation syntax.		Correctly identifies Thread class or Runnable interface but cannot explain the thread lifecycle.		Clearly explains threads, demonstrates both Thread and Runnable approaches, and describes the thread lifecycle.

Detail

Object-Oriented Programming (OOP) Concepts

Score: 66



Description:

Covers the core principles of object-oriented design in Java, including classes, objects, inheritance, polymorphism, encapsulation, and interfaces. These concepts form the foundation of nearly all Java application development and are applied constantly when writing and organizing code.

Interpretation:

Candidate should achieve above average job performance in this area with little or no training.

The candidate exhibits a solid working knowledge of Java programming, including competence in object-oriented principles, collections, exception management, database integration, and input/output handling. This individual is well-suited for entry-level to mid-level Java development roles and can be expected to contribute effectively to software projects with minimal supervision.

Interview Guide

How does inheritance work in Java, and can you describe a situation where you used it to reduce code duplication in a project?

1	2	3	4	5
Defines inheritance vaguely with no practical example or confuses it with other OOP concepts.		Correctly explains inheritance and provides a basic example but misses nuances like method overriding.		Clearly explains inheritance, method overriding, and super keyword with a concrete, relevant project example.

Can you explain what a class is in Java and describe how you would use one in a simple program?

1	2	3	4	5
Vague or incorrect explanation; cannot connect class concept to practical use.		Correctly defines a class and gives a basic example but lacks depth on methods or fields.		Clearly defines a class, explains fields and methods, and connects it to a real coding scenario.

IT Coding Tasks

During the assessment, the candidate was asked to write one or more programs or scripts. Their responses are included below for review.

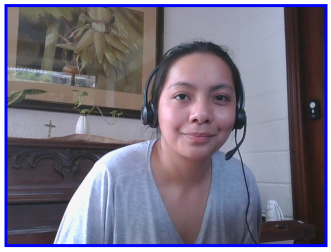
Question or Task	Response

Complete the provided partial C program by filling in the missing sections marked with TODO comments. Your completion must use standard C keywords and libraries.
 	
The program declares a function called duplicate_array that:
1. Takes a const int pointer to a source array and its length as parameters.
2. Uses calloc to allocate a new int array of the same length.
3. Returns NULL if calloc fails.
4. Copies each element from the source array into the new array using pointer arithmetic (not array subscript notation).
5. Returns the pointer to the newly allocated copy.
 	
In main, the program:
1. Declares and initializes a stack array of 4 integers with values 5, 15, 25, 35.
2. Calls duplicate_array to create a heap-allocated copy.
3. Checks for NULL and prints an error and returns 1 if the call failed.
4. Prints each element of the duplicate using a loop.
5. Frees the duplicate array.
 	
Use appropriate indentation, common C coding conventions, and add brief inline comments where needed. Type your completed source code as your response.
	#include <stdio.h>
	#include <stdlib.h>
	
	int *duplicate_array(const int *src, int length) {
	/* TODO: Use calloc to allocate a new array of 'length' integers, return
	NULL if calloc fails, copy elements from src using pointer arithmetic,
	and return the new pointer. */
	calloc(303);
	}
	
	int main(void) {
	/* TODO: Declare and initialize a stack array of 4 integers: 5, 15, 25, 35,
	then call duplicate_array and store the result. Check for NULL and
	print an error message returning 1 if it failed. */
	array[4]={5,15,25,35};
	
	int i;
	
	/* Print each element of the duplicate */
	for (i = 0; i < 4; i++) {
	printf("duplicate[%d] = %d\n", i, *(duplicate + i));
	}
	
	/* Free the duplicate array */
	free(duplicate);
	return 0;
	}
	
	Comments (AI): The code segment has several syntax errors and incomplete implementation. The duplicate_array function does not correctly allocate memory or copy elements. The main function has syntax errors and does not properly call the duplicate_array function. However, the structure and intent of the code are somewhat clear, and the code attempts to follow the requirements.

Identity Confirmation Photos

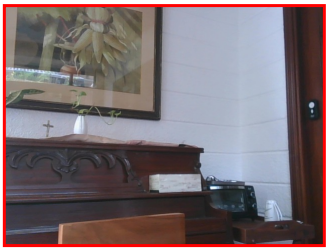
The following photos of the candidate and any identification were uploaded during the assessment session.

Photo Analysis Results	
- Risk:	Medium risk of cheating based on image inconsistencies
- Percent match among processed faces	100%
- Total images processed	17
- Total images with valid faces	14 (82%)
- Total pairs of faces compared	13
- Pairs in which faces matched	13 (100%)

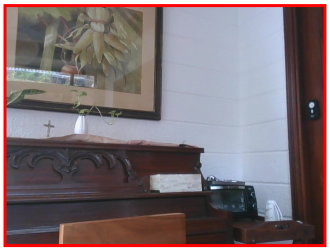


Pre/Post-Test Photo

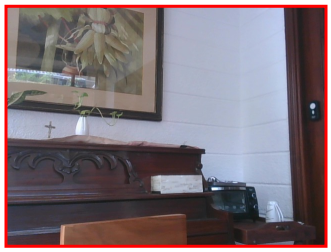
ID Photo



In-Test Error Detected (No Face Detected)

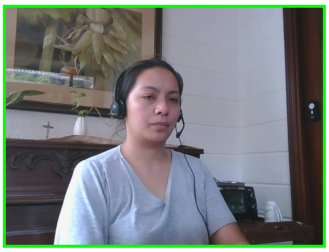


In-Test Error Detected (No Face Detected)

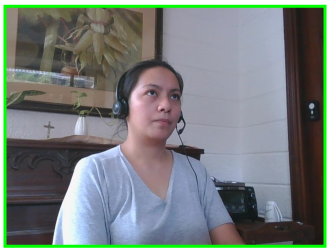


In-Test Error Detected (No Face Detected)

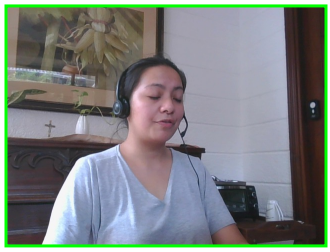
In-Test Photo



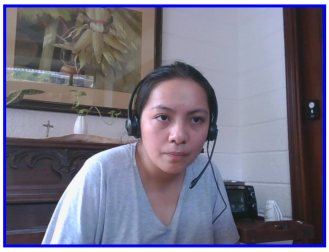
In-Test Photo



In-Test Photo



In-Test Photo



Pre/Post-Test Photo

Resume or CV

Summary Updated on

Motivated career professional with extensive experience in office administration and management. Proven track record of improving efficiency, reducing costs, and enhancing office operations through strategic initiatives and technology implementation.

Objective

I am seeking a role where I can use my many skills and my exceptional judgment and empathy for customers to make a difference to a growing company.

Education

- Associate of Applied Science in Office Administration, Portland Community College, 2020

Experience

- General Office Clerk, Paramount Office Management, 09/2023 – Present
- Administrative Assistant, Global Enterprises Inc., 04/2021 – 08/2023
- Administrative Assistant, Innovative Business Solutions Ltd., 07/2019 – 03/2021

Other Qualifications

- Microsoft Office Specialist (MOS) Certification
- Certified Administrative Professional (CAP)
- International Association of Administrative Professionals (IAAP) Certification

Report Preparation Notes

- Hiring decisions should never be based on a single source of information. The most effective use of this assessment report is as a part of a multi-faceted program of candidate evaluation that includes resume review, interviews, and reference checks.
- Overall vs Percentiles Scores: The overall score reflects the success in the test, based on the mean (average) and standard deviation of the test scores. The percentile score reflects the percentage of test-takers who scored equal or below this overall score. We recommend you use the Overall Score as your primary evaluation criteria. However, percentile scores can often be useful in comparing specific candidates against one another and with a group, such as for test takers in a certain organization or within a certain account.
- Note that comparison information is calculated based on completed instances of this assessment at that time the assessment is scored. As additional instances are completed, the comparative data may change. You can always update a report to the current values by clicking on 'Recalculate Percentiles' within the online results viewing pages at www.hravatar.com.
- Most competency scores are norm-based, which means that they can be interpreted in terms of their distance from the average or mean score. For all scales, a score equal to the mean receives a score of 65 and scores above and below this value are set so that a score change of 15 equals one standard deviation.
- For linear competencies, higher is better across the entire scale. For these scales a score between 65 and 80 (light green) represents 0 to 1 standard deviation above the mean and a score above 80 (dark green) represents more than one standard deviation above the mean. Similarly, a score of 50 - 65 (yellow) represents 0 to 1 standard deviation below the mean, while a score of 35 - 50 (orange) equates to 1 to 2 standard deviations below the mean, and a score below 35 represents more than 2 standard deviations below the mean.
- Sim ID: 20663-1, Key: 0-0, Rpt: 68, Prd: 9541, Created: 2026-06-24 22:57 EDT
- UA: Mozilla/5.0 (Windows NT 6.3; Trident/7.0; Touch; rv:11.0) like Gecko

Score Calculation Detail

The following table provides a summary of how the overall score was calculated from each of the individual competency scores. First, all competency scores are calculated on a scale of 0-100. Note that some competencies use their color category rather than their actual numeric score in the overall calculation. For these, a standard score associated with the assigned color category is used in the overall score calculation rather than the actual numeric score. This is reflected in the ''Score Value Used'' column. Next, a weighted average of scores is computed using individual competency weights, typically set using job analysis data provided by the US Government Occupational Information Network (O*Net).

Competency	Score	How applied to overall	Score Value Used	Weight (%)
Collections Framework and Data Structures	78.4386	Numeric Score	78.4386	12.5000
Collections Framework and Data Structures (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Database Connectivity (JDBC)	68.4650	Numeric Score	68.4650	12.5000
Exception Handling	69.8252	Numeric Score	69.8252	12.5000
Input/Output (I/O) and Data Serialization	68.5532	Numeric Score	68.5532	12.5000
Multithreading and Concurrency	66.2332	Numeric Score	66.2332	12.5000
Object-Oriented Programming (OOP) Concepts	66.1995	Numeric Score	66.1995	12.5000
Object-Oriented Programming (OOP) Concepts (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Weighted Average:				67.9589
Final Overall Score:				67

Notes

(This area is intentionally blank - it's reserved as space for your notes.)