hr | avatar
Illuminating Talent

Test Results and Interview Guide

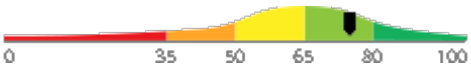
Candidate: **Elizabeth Wantsajob**
Assessment: Windows PowerShell Programming
Completed: June 27, 2026
Prepared for: Sara Maple
Example Company

What's Included

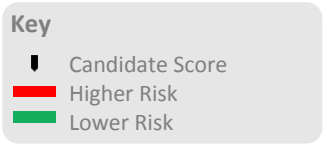
- Overall Score
- Competency Summary Table
- Comparison Matrix
- Detailed Competency Results with Interview Guide

Important Note: The Windows PowerShell 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.

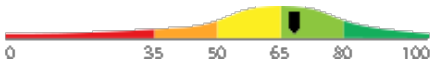
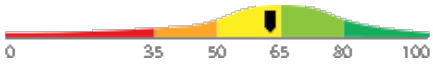
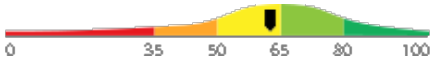
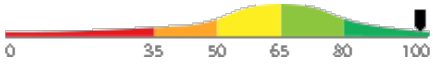
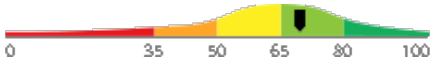
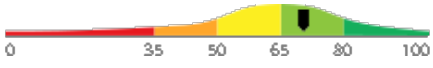
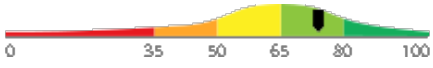
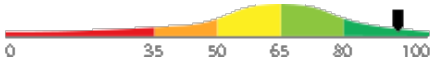
Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com Windows PowerShell Programming June 27, 2026	75	

The candidate exhibits a solid and competent understanding of Windows PowerShell scripting, demonstrating proficiency across a broad range of topics including cmdlet usage, pipeline operations, error handling, file I/O, and object manipulation. They are likely capable of writing, debugging, and maintaining functional scripts for automation and administration tasks with moderate complexity. Minor gaps may exist in advanced areas such as remote execution, registry interaction, or regular expressions, but overall this individual is well-suited for entry-level to mid-level PowerShell scripting responsibilities.



Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Cmdlets, Pipeline, and Output Formatting	68	
Cmdlets, Pipeline, and Output Formatting (Coding Tasks)	62	
Variables, Data Types, and Operators (Coding Tasks)	62	
Control Flow: Conditionals and Loops	98	
Error Handling and Debugging	69	
File Input/Output and Data Formats	70	
Functions, Parameters, and Script Structure	74	
Variables, Data Types, and Operators	93	

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	75th
North America	62nd
United States	62nd
Example Company	69th

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:	Windows PowerShell Programming
Authorized:	June 27, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
Started:	June 27, 2026, 2:30:37PM EDT
Completed:	June 27, 2026, 2:30:37PM EDT
Overall Score:	75

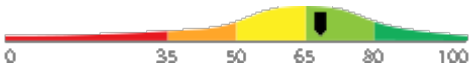
Knowledge and Skills Detail

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

Cmdlets, Pipeline, and Output Formatting

Score: 68



Description:

Covers the use of built-in PowerShell cmdlets to perform common tasks such as file management, process handling, and string manipulation. Includes how to chain commands together using the pipeline, and how to filter, sort, and format output using cmdlets like Where-Object, Sort-Object, Select-Object, and Format-Table.

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 Windows PowerShell scripting, including proficiency with core syntax, control structures, cmdlets, error handling, file I/O, and object manipulation. They are well-equipped to independently write, debug, and maintain scripts for a variety of automation and system administration tasks, with only occasional gaps in more advanced or specialized areas.

Walk me through how you would retrieve all files in a directory that are larger than 10MB, display only their names and sizes, and sort the results by size in descending order using a single PowerShell pipeline.

1	2	3	4	5
Cannot construct the pipeline or uses incorrect cmdlets and syntax.		Identifies the correct cmdlets but pipeline is incomplete or sort order is missing or wrong.		Correctly chains Get-ChildItem, Where-Object, Select-Object, and Sort-Object with proper parameters and descending sort.

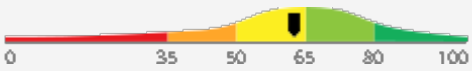
Can you explain what the PowerShell pipeline does and give a simple example of how you would use it to filter a list of running processes by name?

1	2	3	4	5
Cannot explain the pipeline or provides an unrelated or incorrect example.		Describes the pipeline generally but example is incomplete or contains minor errors.		Clearly explains piping output between cmdlets and gives a correct, specific example using Get-Process and Where-Object.

Detail

Cmdlets, Pipeline, and Output Formatting (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.

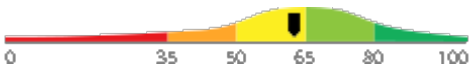
Can you explain what a pointer is in C and describe a situation where you would use one?

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

Variables, Data Types, and Operators (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.

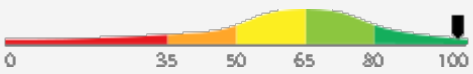
Can you explain what a pointer is in C and describe a situation where you would use one?

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

Control Flow: Conditionals and Loops

Score: 98



Description:

Covers the use of conditional statements such as if, elseif, and else, as well as loop structures including foreach, for, while, and do-while. Includes understanding of how to control script execution based on conditions and how to iterate over collections of data to automate repetitive tasks.

Interpretation:

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

The candidate demonstrates a comprehensive and advanced understanding of PowerShell control flow, including conditional statements and all major loop structures. They are highly proficient in controlling script execution based on complex conditions and efficiently automating repetitive tasks by iterating over data collections.

Interview Guide

How would you write a PowerShell script that reads a list of user account names and, for each one, checks whether the account is enabled or disabled and takes a different action depending on the result?

1	2	3	4	5
Cannot combine a loop with conditional logic or does not know relevant cmdlets for checking account status.		Correctly structures the loop and conditional but cmdlet usage or property references are incomplete or slightly incorrect.		Correctly uses a foreach loop, Get-ADUser or similar cmdlet, and if/else logic with accurate property checks and distinct actions per condition.

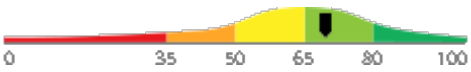
Can you write a simple PowerShell script that loops through a list of five names and prints a greeting for each one?

1	2	3	4	5
Cannot write a working loop or does not know how to iterate over a collection.		Writes a loop that mostly works but contains syntax errors or does not correctly reference the loop variable.		Writes a clean foreach loop over an array of names and correctly outputs a personalized greeting for each.

Detail

Error Handling and Debugging

Score: 69



Description:

Covers how to handle errors in PowerShell scripts using try, catch, and finally blocks, and how to work with terminating and non-terminating errors. Includes use of $ErrorActionPreference, the -ErrorAction parameter, Write-Debug, and breakpoints for diagnosing and fixing problems in scripts.

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 PowerShell error handling and debugging practices. They are likely proficient with try, catch, and finally blocks, error action preferences, and common debugging tools, with only minor gaps in more advanced or nuanced areas.

Interview Guide

How would you debug a PowerShell script that is producing unexpected output? Walk me through the tools and techniques you would use to identify and fix the problem.

1	2	3	4	5
Can only suggest re-reading the code manually with no knowledge of PowerShell debugging tools.		Mentions Write-Host or Write-Debug for tracing but does not mention breakpoints or the ISE/VS Code debugger.		Describes a systematic approach using Write-Debug, Set-PSBreakpoint or IDE breakpoints, $DebugPreference, and stepping through code to isolate the issue.

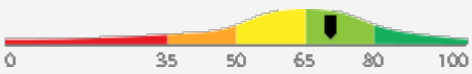
What happens when an error occurs in a PowerShell script, and how would you use a try-catch block to prevent the script from stopping and instead display a helpful error message?

1	2	3	4	5
Cannot describe error handling or does not know the structure of a try-catch block.		Describes try-catch generally and provides a partial example but syntax or error message output is incomplete.		Correctly explains terminating errors, writes a try-catch with a specific action in the catch block, and references $_.Exception.Message or similar.

Detail

File Input/Output and Data Formats

Score: 70



Description:

Covers reading from and writing to files in common formats including plain text, CSV, and JSON. Includes use of cmdlets such as Get-Content, Set-Content, Import-Csv, Export-Csv, ConvertTo-Json, and ConvertFrom-Json to handle data as part of automation and administration scripts.

Interpretation:

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

The candidate demonstrates a solid working knowledge of file input/output operations and data format handling in Windows PowerShell. They are generally proficient with key cmdlets for managing plain text, CSV, and JSON data within automation and administration scripts.

Interview Guide

How would you import data from a CSV file in PowerShell, filter the rows based on a specific column value, and then export the filtered results to a new CSV file?

1	2	3	4	5
Cannot identify the correct cmdlets for importing or exporting CSV data.		Correctly uses Import-Csv and Export-Csv but the filtering step is missing or uses incorrect syntax.		Correctly chains Import-Csv, Where-Object with an accurate column reference, and Export-Csv with -NoTypeInformation in a clean pipeline.

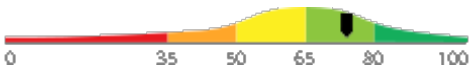
How would you read the contents of a text file in PowerShell and display each line on the screen?

1	2	3	4	5
Does not know how to read a file or cannot name the correct cmdlet.		Correctly identifies Get-Content but does not explain how to iterate over lines or display them clearly.		Correctly uses Get-Content, explains that it returns an array of lines, and demonstrates iterating or piping the output to display each line.

Detail

Functions, Parameters, and Script Structure

Score: 74



Description:

Covers how to define and call reusable functions, declare parameters with appropriate data types and validation, and structure scripts for readability and maintainability. Includes use of param() blocks, parameter attributes such as

Interpretation:

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

The candidate demonstrates solid proficiency in defining and calling reusable PowerShell functions, declaring parameters with appropriate data types, and structuring scripts for readability and maintainability. They are generally competent with parameter attributes such as mandatory designations and default values, though mastery of advanced or edge-case scenarios may still be developing.

Interview Guide

How would you write a PowerShell function that requires a mandatory string parameter, validates that the input is not empty, and includes a comment-based help block so other users know how to use it?

1	2	3	4	5
Cannot add parameter validation or does not know how to write comment-based help.		Adds a mandatory parameter and some documentation but validation or help block syntax is incomplete.		Correctly uses [Parameter(Mandatory)], [ValidateNotNullOrEmpty()], and a complete .SYNOPSIS/.DESCRIPTION help block with accurate syntax.

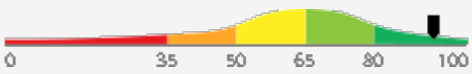
How do you define a function in PowerShell, and how would you pass a value into it so the function can use that value in its logic?

1	2	3	4	5
Cannot correctly define a function or does not know how to declare or use a parameter.		Correctly defines a function and passes a parameter but syntax or usage contains minor errors.		Clearly defines a function with the function keyword, declares a named parameter, and correctly references it inside the function body.

Detail

Variables, Data Types, and Operators

Score: 93



Description:

Covers how to declare and use variables, work with common data types such as strings, integers, arrays, and hashtables, and apply arithmetic, comparison, and logical operators. Includes understanding of how PowerShell handles type conversion and how to use variables effectively within scripts.

Interpretation:

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

The candidate demonstrates a comprehensive and advanced understanding of PowerShell variables, data types, and operators. They are highly proficient in declaring and using variables, working across all common data types, applying operators accurately, and managing type conversion effectively within scripts.

Interview Guide

How would you use a hashtable in PowerShell to store a set of key-value pairs, and how would you access, add, and remove entries from it within a script?

1	2	3	4	5
Cannot correctly define or interact with a hashtable, or confuses it with an array.		Correctly creates a hashtable and accesses values but struggles with adding or removing entries.		Accurately demonstrates hashtable creation with @{}, key access, adding new keys, and using .Remove() with clear syntax.

How do you create a variable in PowerShell and store a number in it? What would you do to add two variables together and display the result?

1	2	3	4	5
Cannot correctly declare a variable or describe basic arithmetic operations in PowerShell.		Correctly declares a variable and performs addition but explanation lacks detail or contains minor syntax errors.		Clearly explains variable declaration with $, assigns numeric values, and correctly demonstrates addition and output with Write-Host or similar.

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.

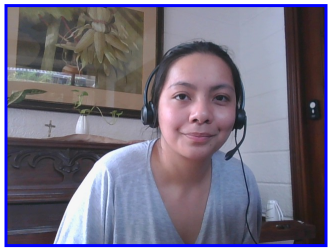
Question or Task	Response

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

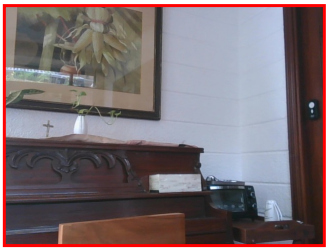
The following photos of the candidate and any identification were uploaded during the assessment session.

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%)

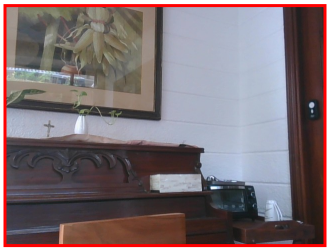


Pre/Post-Test Photo

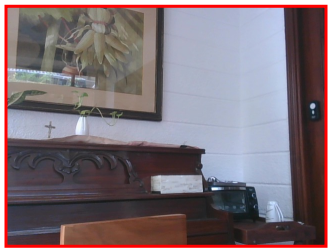
ID Photo



In-Test Error Detected (No Face Detected)

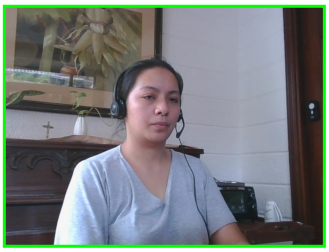


In-Test Error Detected (No Face Detected)

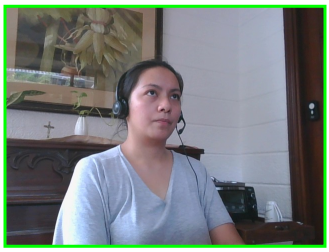


In-Test Error Detected (No Face Detected)

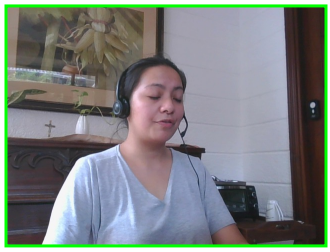
In-Test Photo



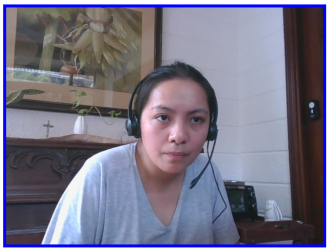
In-Test Photo



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: 20795-1, Key: 0-0, Rpt: 68, Prd: 9617, Created: 2026-06-27 14:30 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 (%)
Cmdlets, Pipeline, and Output Formatting	68.4651	Numeric Score	68.4651	12.5000
Cmdlets, Pipeline, and Output Formatting (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Control Flow: Conditionals and Loops	98.5387	Numeric Score	98.5387	12.5000
Error Handling and Debugging	69.8062	Numeric Score	69.8062	12.5000
File Input/Output and Data Formats	70.7433	Numeric Score	70.7433	12.5000
Functions, Parameters, and Script Structure	74.4565	Numeric Score	74.4565	12.5000
Variables, Data Types, and Operators	93.0390	Numeric Score	93.0390	12.5000
Variables, Data Types, and Operators (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Weighted Average:				75.1257
Final Overall Score:				75

Notes

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