hr | avatar
Illuminating Talent

Test Results and Interview Guide

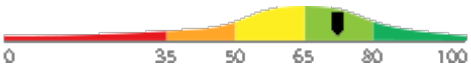
Candidate: **Elizabeth Wantsajob**
Assessment: JSON Syntax and Usage (Short)
Completed: June 28, 2026
Prepared for: Sara Maple
Example Company

What's Included

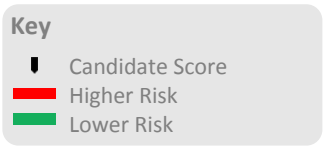
- Overall Score
- Competency Summary Table
- Comparison Matrix
- Detailed Competency Results with Interview Guide

Important Note: The JSON Syntax and Usage (Short) 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.

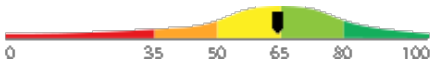
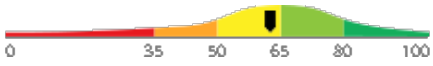
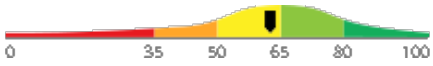
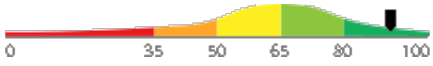
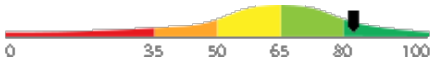
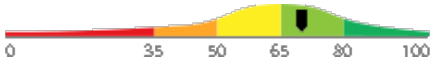
Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com JSON Syntax and Usage (Short) June 28, 2026	72	

The candidate exhibits a solid and competent understanding of JSON syntax and usage, including the ability to construct and interpret objects and arrays, apply syntax rules accurately, and identify and correct common errors in JSON documents. This individual is well-prepared to work with JSON in typical business application contexts, such as APIs and configuration files, with only occasional need for guidance on more advanced topics.



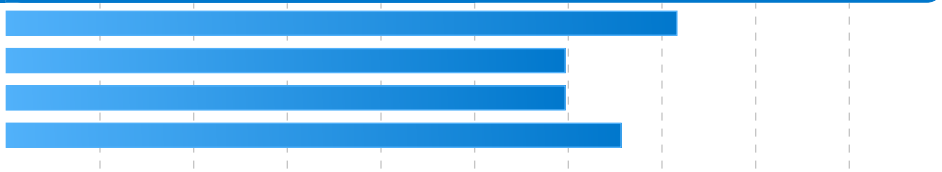
Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
JSON Data Types	64	
JSON Data Types (Coding Tasks)	62	
JSON Syntax Rules (Coding Tasks)	62	
JSON Syntax Rules	91	
Key-Value Pair Structure	82	
Nested Objects and Arrays	70	

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	72nd
North America	60th
United States	60th
Example Company	66th



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:	JSON Syntax and Usage (Short)
Authorized:	June 28, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
Started:	June 28, 2026, 12:13:55PM EDT
Completed:	June 28, 2026, 12:13:55PM EDT
Overall Score:	72

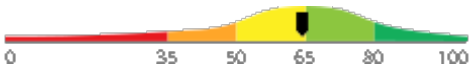
Knowledge and Skills Detail

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

JSON Data Types

Score: 64



Description:

JSON supports six core data types: strings, numbers, booleans, arrays, objects, and null. Understanding how each data type is written and used correctly is essential for constructing and reading any JSON document.

Interpretation:

Candidate appears capable of average job performance in this area with little or no training.

The candidate possesses a partial understanding of JSON syntax and usage, reflecting competence in some fundamental areas while exhibiting inconsistency in more advanced topics such as nesting, escape character encoding, and error identification. With targeted training and practical experience, this individual has the potential to develop into a proficient contributor in roles requiring JSON knowledge.

In a JSON document representing a user profile, how would you decide whether to represent a field like 'is_active' as a boolean versus a string, and what are the practical consequences of that choice?

1	2	3	4	5
Cannot clearly distinguish between the two types or explain any practical consequences.		Explains the difference but gives only a vague or partial account of practical consequences.		Clearly explains the distinction and describes concrete consequences such as parsing behavior or type-checking in code.

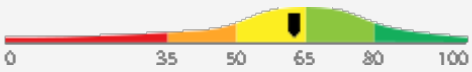
Can you name the data types that JSON supports and give a simple example of each?

1	2	3	4	5
Names fewer than three data types or provides mostly incorrect examples.		Names most data types but struggles to give correct examples for all of them.		Correctly names all six data types and provides a clear, accurate example of each.

Detail

JSON Data Types (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.

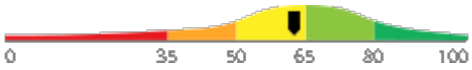
Can you explain what a pointer is in C and describe a situation where you would use one?

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

JSON Syntax Rules (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.

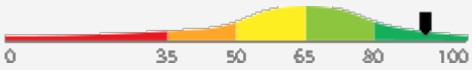
Can you explain what a pointer is in C and describe a situation where you would use one?

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

JSON Syntax Rules

Score: 91



Description:

JSON has strict rules for how data must be written, including the use of curly braces for objects, square brackets for arrays, colons to separate keys and values, and commas to separate items. Knowing these rules is necessary to write valid JSON that can be read by any system.

Interpretation:

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

The candidate demonstrates a comprehensive and thorough understanding of JSON syntax rules, including the precise and correct application of all structural elements required to produce valid JSON. They are highly proficient in writing JSON that adheres to strict formatting standards and can be reliably interpreted by any system.

Interview Guide

What is the most common syntax mistake you have seen or made in a JSON document, and how did you find and fix it?

1	2	3	4	5
Cannot describe a specific syntax mistake or explain how to identify and correct it.		Describes a common mistake such as a trailing comma but gives only a general explanation of how to fix it.		Describes a specific, realistic mistake, explains clearly why it breaks JSON validity, and outlines a precise correction process.

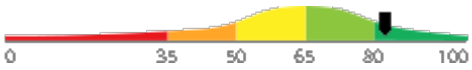
If you were writing a JSON object with two key-value pairs, what punctuation would you use and where would you place it?

1	2	3	4	5
Cannot correctly describe the required punctuation or places it incorrectly.		Describes most of the required punctuation correctly but omits or misplaces one element.		Accurately describes all required punctuation, including braces, colon, and comma, and places each correctly.

Detail

Key-Value Pair Structure

Score: 82



Description:

JSON objects are built from key-value pairs, where every key must be a string enclosed in double quotes, followed by a colon and then a value. Understanding how to correctly form, name, and use key-value pairs is fundamental to working with JSON objects.

Interpretation:

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

The candidate exhibits a strong, comprehensive understanding of JSON key-value pair structure, including precise application of syntax rules such as double-quoted string keys, colon separators, and valid value usage. They are well-equipped to work confidently and accurately with JSON objects in professional settings.

Interview Guide

How would you design the key-value pairs in a JSON object to represent a product in an online store, and what naming practices would you follow to keep the structure clear and consistent?

1	2	3	4	5
Proposes a vague or poorly structured design with no consideration of naming consistency.		Proposes a reasonable structure but applies naming conventions inconsistently or without explanation.		Proposes a clear, well-structured design and explains consistent naming conventions such as camelCase or snake_case with a rationale.

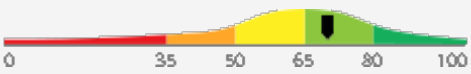
What are the rules for writing a key in a JSON object, and what happens if you break one of those rules?

1	2	3	4	5
Cannot state that keys must be strings in double quotes or cannot describe any consequence of breaking the rules.		States that keys must be quoted strings but cannot clearly explain consequences of invalid key formatting.		Accurately states all key rules including double-quote requirement and explains that violations cause a parse error.

Detail

Nested Objects and Arrays

Score: 70



Description:

JSON allows objects and arrays to be placed inside other objects and arrays, which makes it possible to represent complex, real-world data relationships. Knowing how to build and read nested structures is critical for working with APIs and configuration files.

Interpretation:

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

The candidate exhibits a solid understanding of nested objects and arrays in JSON, and is generally capable of building and reading complex nested structures. Minor gaps in knowledge may exist, but they are well-positioned to work effectively with APIs and configuration files in most practical scenarios.

Interview Guide

When reading a deeply nested JSON structure from an API response, what approach do you take to understand the data and access the values you need?

1	2	3	4	5
Cannot describe a clear or practical approach to navigating nested JSON.		Describes a basic approach such as reading the structure visually but does not mention tools or programmatic access methods.		Describes a systematic approach including use of tools, dot or bracket notation, or code-based traversal to locate and extract values.

How would you write a JSON object that contains a list of items, where each item in the list also has its own properties?

1	2	3	4	5
Cannot describe or write a structure that correctly nests an array of objects.		Describes the general idea but makes syntax errors or cannot clearly explain how nesting works.		Correctly describes or writes a nested array of objects with accurate syntax and clear explanation.

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.

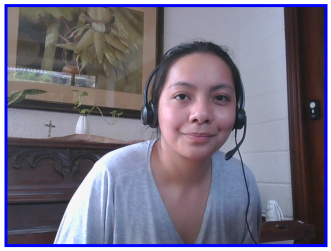
Question or Task	Response

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

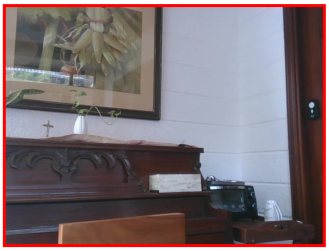
The following photos of the candidate and any identification were uploaded during the assessment session.

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%)

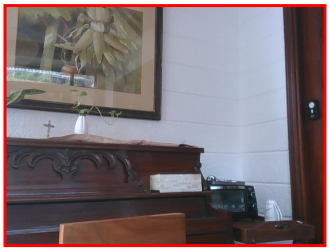


Pre/Post-Test Photo

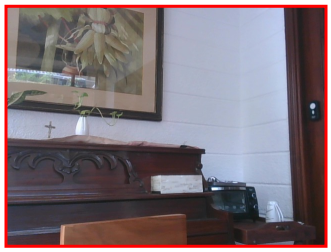
ID Photo



In-Test Error Detected (No Face Detected)

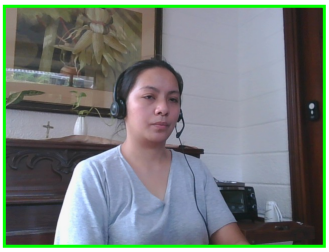


In-Test Error Detected (No Face Detected)

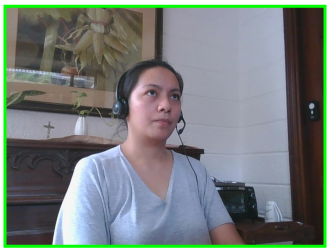


In-Test Error Detected (No Face Detected)

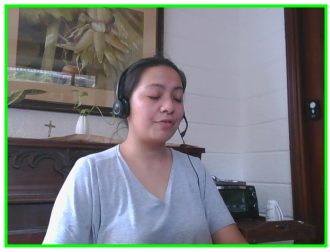
In-Test Photo



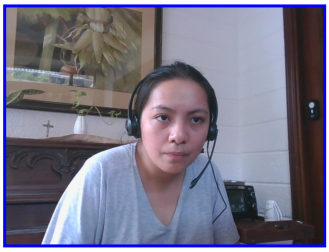
In-Test Photo



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: 20813-1, Key: 0-0, Rpt: 68, Prd: 9635, Created: 2026-06-28 12:13 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 (%)
JSON Data Types	64.8639	Numeric Score	64.8639	16.6667
JSON Data Types (Coding Tasks)	62.9784	Numeric Score	62.9784	16.6667
JSON Syntax Rules	91.6652	Numeric Score	91.6652	16.6667
JSON Syntax Rules (Coding Tasks)	62.9784	Numeric Score	62.9784	16.6667
Key-Value Pair Structure	82.5158	Numeric Score	82.5158	16.6667
Nested Objects and Arrays	70.2351	Numeric Score	70.2351	16.6667
Weighted Average:				72.5395
Final Overall Score:				72

Notes

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