hr | avatar
Illuminating Talent

Test Results and Interview Guide

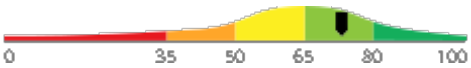
Candidate: **Elizabeth Wantsajob**
Assessment: XML Syntax and Usage
Completed: June 28, 2026
Prepared for: Sara Maple
Example Company

What's Included

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

Important Note: The XML Syntax and Usage 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.

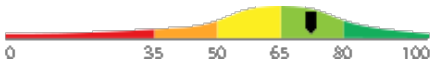
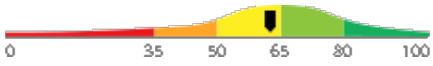
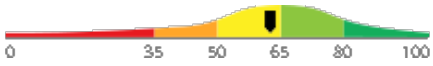
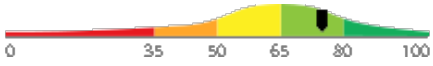
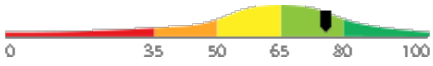
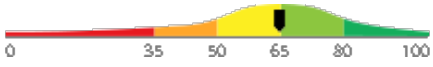
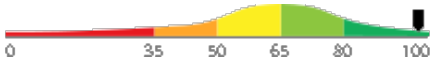
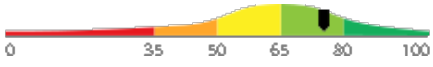
Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com XML Syntax and Usage June 28, 2026	73	

The candidate demonstrates a solid working knowledge of XML syntax and its application in business contexts, including document structure definition, validation, and transformation using standard technologies. Minor gaps may exist in specialized areas such as advanced schema design, encoding considerations, or the nuanced use of parsing models, but overall competence is evident.

Key
- Candidate Score
- Higher Risk
- Lower Risk

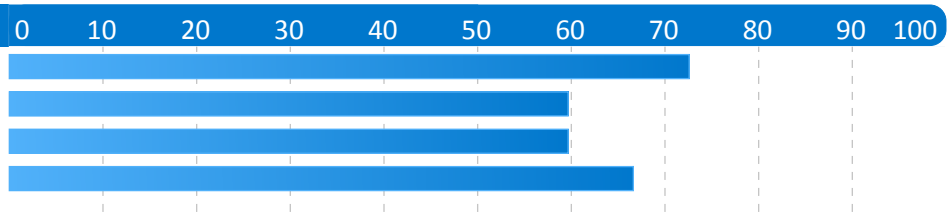
Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Character Encoding and Special Characters	72	
XML Document Structure and Syntax Rules (Coding Tasks)	62	
XML Schema and DTD for Document Validation (Coding Tasks)	62	
Namespaces in XML	75	
Parsing XML with DOM and SAX	76	
XML Document Structure and Syntax Rules	65	
XML Schema and DTD for Document Validation	98	
XPath and XSLT for Navigating and Transforming XML	75	

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	73rd
North America	60th
United States	60th
Example Company	67th



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:	XML Syntax and Usage
Authorized:	June 28, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
Started:	June 28, 2026, 10:12:57AM EDT
Completed:	June 28, 2026, 10:12:57AM EDT
Overall Score:	73

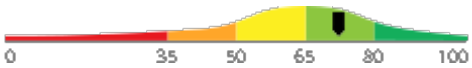
Knowledge and Skills Detail

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

Character Encoding and Special Characters

Score: 72



Description:

Covers how XML handles text encoding, including the declaration of character sets such as UTF-8 and UTF-16, and the correct use of predefined entity references and CDATA sections to include special characters that would otherwise break XML syntax.

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 XML character encoding and special character handling. They can reliably work with character set declarations, predefined entity references, and CDATA sections, with only minor gaps that are unlikely to significantly impact their performance in most professional contexts.

Why is it important to declare the character encoding in an XML document, and what issues can occur when encoding is missing or set incorrectly?

1	2	3	4	5
Cannot explain why encoding matters or what problems incorrect encoding causes.		Understands encoding is important but gives only a vague explanation of potential issues.		Clearly explains encoding declaration purpose and describes specific issues like garbled characters or parse errors.

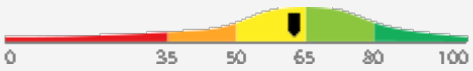
What would you do if you needed to include a special character like an ampersand or a less-than sign inside the content of an XML element?

1	2	3	4	5
Does not know how to handle special characters or suggests an approach that would break XML.		Knows that special handling is needed but can only recall one or two entity references.		Accurately describes entity references and CDATA sections as options, with correct syntax examples.

Detail

XML Document Structure and 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.

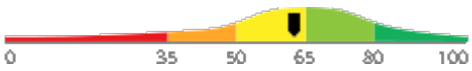
Can you explain what a pointer is in C and describe a situation where you would use one?

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

XML Schema and DTD for Document Validation (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.

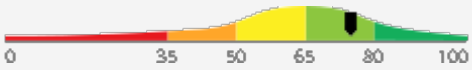
Can you explain what a pointer is in C and describe a situation where you would use one?

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

Namespaces in XML

Score: 75



Description:

Covers the use of XML namespaces to avoid naming conflicts when combining XML vocabularies from different sources, such as when integrating data from multiple systems or using standard XML-based formats in business applications.

Interpretation:

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

The candidate exhibits a solid understanding of XML namespaces, including their practical application in avoiding naming conflicts when integrating data from multiple systems or working with standard XML-based formats. They are well-equipped to handle most namespace-related tasks, with only occasional difficulty in advanced or edge-case scenarios.

Interview Guide

How do you declare and use a namespace in an XML document, and what problems can arise if namespaces are used incorrectly or inconsistently?

1	2	3	4	5
Cannot describe the declaration syntax or identify any problems with incorrect namespace use.		Describes the declaration generally but is vague about problems caused by inconsistent use.		Accurately describes xmlns declaration and prefix use, and identifies specific problems like element name conflicts.

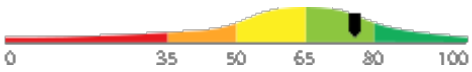
Can you explain what a namespace is in XML and describe a situation where using one would be necessary?

1	2	3	4	5
Cannot explain namespaces or gives a response that confuses them with another concept.		Gives a basic definition but cannot describe a realistic scenario where namespaces are needed.		Clearly defines namespaces and describes a concrete scenario such as merging XML from two systems.

Detail

Parsing XML with DOM and SAX

Score: 76



Description:

Covers how to read and work with XML data in code using the Document Object Model (DOM) and Simple API for XML (SAX) parsing approaches. This includes understanding when to use each approach and how to access or manipulate XML content programmatically.

Interpretation:

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

The candidate demonstrates a solid understanding of both DOM and SAX parsing approaches and is capable of reading, accessing, and manipulating XML content programmatically in most scenarios. They show competence in selecting the appropriate parsing method based on context, with only minor gaps in more advanced application.

Interview Guide

What are the key differences between DOM and SAX parsing, and how would you decide which one to use when processing a large XML file in a business application?

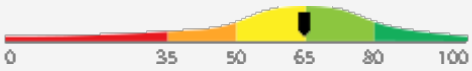
1	2	3	4	5
Cannot distinguish DOM from SAX or gives inaccurate descriptions of both.		Describes the general difference but cannot clearly connect the choice to performance or memory concerns.		Accurately contrasts DOM and SAX and gives a well-reasoned recommendation based on file size and use case.

Can you explain what it means to parse an XML document and describe one way a program might read data out of an XML file?

1	2	3	4	5
Cannot explain parsing or gives a response unrelated to XML processing.		Explains parsing generally but cannot describe a specific method or approach in XML.		Clearly explains parsing and accurately describes DOM or SAX with a practical example.

XML Document Structure and Syntax Rules

Score: 65



Description:

Covers the core rules for writing well-formed XML documents, including proper use of elements, tags, attributes, nesting, and the XML declaration. This includes understanding how to structure a document so that it can be read and processed by any XML-aware application.

Interpretation:

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

The candidate demonstrates a solid and broadly competent understanding of XML syntax, document structure, validation, and business usage, reflecting knowledge consistent with a capable entry-level to mid-level practitioner. Minor gaps may exist in specialized areas such as advanced schema design, encoding considerations, or complex transformation logic, but overall proficiency is sufficient for most standard XML-related responsibilities.

How do you decide when to represent a piece of data as an XML element versus as an attribute on an element, and why does that distinction matter in practice?

1	2	3	4	5
Cannot articulate a meaningful difference or gives only a superficial answer.		Explains basic difference but does not connect it to practical or interoperability concerns.		Clearly explains tradeoffs with practical examples, mentioning readability, data complexity, and reuse.

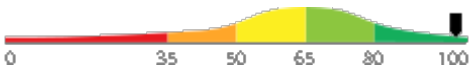
Can you walk me through what makes an XML document 'well-formed' and give an example of a common syntax mistake you might encounter?

1	2	3	4	5
Cannot define well-formed; struggles to identify basic syntax errors.		Defines well-formed correctly but gives vague or incomplete examples of errors.		Clearly defines well-formed, gives specific examples like unclosed tags or improper nesting.

Detail	Interview Guide

XML Schema and DTD for Document Validation

Score: 98



Description:

Covers how to define the allowed structure, elements, attributes, and data types of an XML document using Document Type Definitions (DTDs) and XML Schema Definition (XSD). This includes writing and applying schemas to validate that documents conform to a required format.

Interpretation:

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

The candidate demonstrates a comprehensive and advanced mastery of XML Schema (XSD) and Document Type Definitions (DTDs) for document validation. They are highly proficient in defining complex document structures, data types, and constraints, and can confidently write and apply schemas to ensure XML documents conform to required formats.

What are the main advantages of using XML Schema (XSD) over a DTD when defining the structure of an XML document used in a business data exchange?

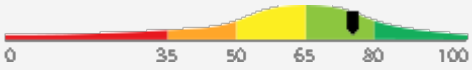
1	2	3	4	5
Cannot identify meaningful differences or advantages of XSD over DTD.		Mentions data types or namespace support but cannot elaborate with practical context.		Clearly contrasts XSD and DTD, citing data types, namespace support, and real-world applicability.

Can you explain what a schema or DTD is used for in XML, and describe what happens when an XML document is validated against one?

1	2	3	4	5
Cannot explain the purpose of a schema or DTD, or confuses validation with parsing.		Explains the general purpose but is unclear on what validation actually checks or produces.		Clearly explains purpose, describes the validation process, and mentions valid vs. invalid document outcomes.

XPath and XSLT for Navigating and Transforming XML

Score: 75



Description:

Covers the use of XPath expressions to locate and select specific elements or data within an XML document, and the use of XSLT stylesheets to transform XML data into other formats such as HTML, plain text, or a different XML structure.

Interpretation:

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

The candidate exhibits a solid and proficient command of XPath expressions and XSLT stylesheets, capable of navigating XML documents and transforming XML data into various output formats with confidence. Minor gaps may exist in edge cases or advanced features, but overall competency is well established for most professional contexts.

Describe a situation where you would use XSLT to solve a real business problem, and walk through the basic steps of how the transformation would work.

1	2	3	4	5
Cannot describe a realistic use case or how XSLT works at a functional level.		Describes a plausible use case but gives only a high-level or incomplete explanation of the process.		Describes a concrete use case and clearly explains template matching, input, and output transformation steps.

Can you describe what XPath is and give an example of how you might use it to find a specific piece of data inside an XML document?

1	2	3	4	5
Cannot describe XPath or provides a completely inaccurate description.		Gives a general description of XPath but cannot provide a usable example expression.		Accurately describes XPath and provides a clear, correct example expression with 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.

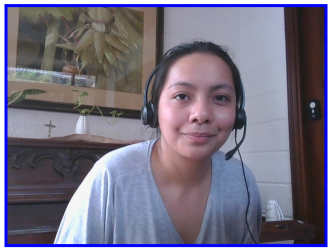
Question or Task	Response

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

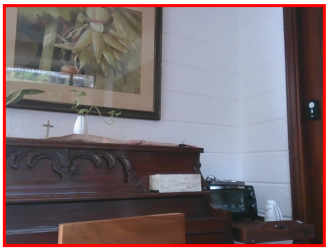
The following photos of the candidate and any identification were uploaded during the assessment session.

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%)

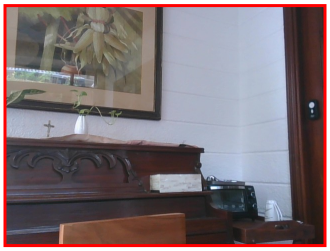


Pre/Post-Test Photo

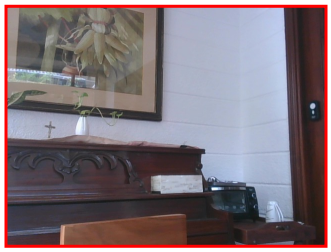
ID Photo



In-Test Error Detected (No Face Detected)

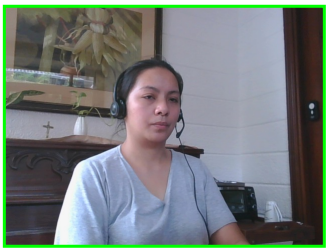


In-Test Error Detected (No Face Detected)

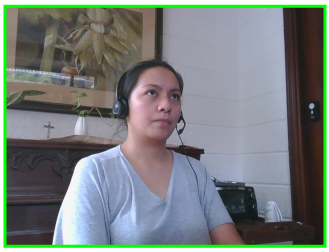


In-Test Error Detected (No Face Detected)

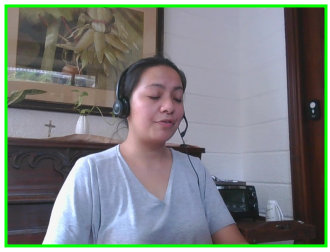
In-Test Photo



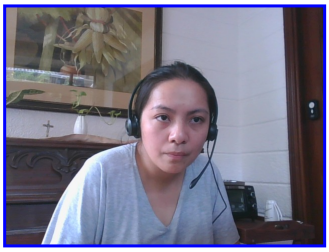
In-Test Photo



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: 20807-1, Key: 0-0, Rpt: 68, Prd: 9629, Created: 2026-06-28 10:12 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 (%)
Character Encoding and Special Characters	72.5913	Numeric Score	72.5913	12.5000
Namespaces in XML	75.3855	Numeric Score	75.3855	12.5000
Parsing XML with DOM and SAX	76.0683	Numeric Score	76.0683	12.5000
XML Document Structure and Syntax Rules	65.0147	Numeric Score	65.0147	12.5000
XML Document Structure and Syntax Rules (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
XML Schema and DTD for Document Validation	98.2191	Numeric Score	98.2191	12.5000
XML Schema and DTD for Document Validation (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
XPath and XSLT for Navigating and Transforming XML	75.4540	Numeric Score	75.4540	12.5000
Weighted Average:				73.5862
Final Overall Score:				73

Notes

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