

Test Results and Interview Guide

Candidate: **Elizabeth Wantsajob**
Assessment: Java Jakarta EE Programming
Completed: June 25, 2026
Prepared for: Sara Maple
Example Company

What's Included

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

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

Overall

Candidate	Score	Interpretation
Elizabeth Wantsajob beth.wantsajob@gmail.com Java Jakarta EE Programming June 25, 2026	68	

The candidate exhibits a solid and well-rounded knowledge of Java Jakarta EE, reflecting competency across most key areas including servlet and JSP development, CDI, JPA-based persistence, RESTful API design, exception handling, and application deployment. Minor gaps may exist in specialized or advanced topics, but the candidate is well-equipped to independently handle the majority of entry-level to mid-level Jakarta EE development responsibilities.

Key

- Candidate Score
- Higher Risk
- Lower Risk

Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Bean Validation and Exception Handling	64	
JAX-RS and RESTful API Development (Coding Tasks)	62	
Servlets, JSP, and HTTP Request/Response Handling (Coding Tasks)	62	
CDI and Dependency Injection	77	
JAX-RS and RESTful API Development	69	
JPA and Database Persistence	67	
Servlets, JSP, and HTTP Request/Response Handling	81	
Transaction Management and Data Access Patterns	64	

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	0	10	20	30	40	50	60	70	80	90	100	
Global	68th												
North America	56th												
United States	56th												
Example Company	63rd												

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	-	-	<p>Summary Points (AI):</p> <ul style="list-style-type: none"> (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. <p>Narrative (AI): Elizabeth Wantsajob demonstrates a mixed profile of knowledge, skills, and abilities across the assessed competencies.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Computed on: April 2, 2026, 11:09:49PM EDT</p>

Detail

Candidate: Elizabeth Wantsajob, beth.wantsajob@gmail.com
 Assessment: Java Jakarta EE Programming
 Authorized: June 25, 2026, by Sara Maple, Example Company, qamailsaram.mike@hravatar.com
 Started: June 25, 2026, 2:24:11PM EDT
 Completed: June 25, 2026, 2:24:11PM EDT
 Overall Score: 68

Knowledge and Skills Detail

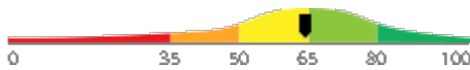
This section contains a list of job-related knowledge areas and skills that have been evaluated. Low scores in these areas often indicate that additional learning may be required before top performance can be achieved.

Detail

Interview Guide

Bean Validation and Exception Handling

Score: 64



Description:

Covers the use of Jakarta Bean Validation constraint annotations to validate input data on beans and method parameters. Also includes strategies for catching, wrapping, and reporting exceptions across the application, including the use of ExceptionMappers in JAX-RS and meaningful error responses.

Interpretation:

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

The candidate possesses a foundational understanding of Jakarta Bean Validation and exception handling principles. They are likely familiar with common constraint annotations and basic exception catching strategies, but may lack consistency or depth when applying more advanced techniques such as exception wrapping or structured error responses in JAX-RS.

Describe how you would implement centralized error handling in a JAX-RS application so that unhandled exceptions are converted into consistent, user-friendly JSON error responses rather than raw stack traces.



1

Has no strategy beyond try-catch in every method; unaware of ExceptionMapper.



2

Knows ExceptionMapper exists but can only implement a basic example without addressing multiple exception types or response structure.



3



4



5

Implements ExceptionMapper hierarchy, maps specific and generic exceptions, returns structured JSON with status codes and messages.

How would you ensure that a field on a Java object, such as a user's email address, is not blank and follows a valid format before it is saved to the database in a Jakarta EE application?



1

Relies only on manual if-checks with no knowledge of Bean Validation annotations.



2

Uses @NotBlank and @Email correctly but cannot explain how validation is triggered or how to handle ConstraintViolationException.



3



4

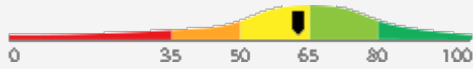


5

Applies constraint annotations, explains automatic triggering via CDI/JAX-RS, and handles violations with clear error messages.

Detail
Interview Guide
JAX-RS and RESTful API Development (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.

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

Cannot write correct allocation code; unaware of free() or memory leak risks.



2

Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.



3



4

Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.



5

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



1

Vague or incorrect definition; cannot describe a practical use case.



2

Correct basic definition; gives a simple but valid use case with some gaps.



3



4

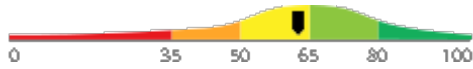
Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.



5

Detail
Interview Guide
Servlets, JSP, and HTTP Request/Response Handling (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.

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

Cannot write correct allocation code; unaware of free() or memory leak risks.



2

Writes mostly correct malloc/free code; identifies memory leaks but misses other risks.



3



4

Correct malloc, use, and free; identifies leaks, dangling pointers, and NULL check on allocation.



5

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



1

Vague or incorrect definition; cannot describe a practical use case.



2

Correct basic definition; gives a simple but valid use case with some gaps.



3



4

Clear definition with accurate use case; mentions address storage, dereferencing, or dynamic memory.

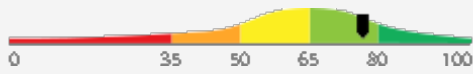


5

Detail Interview Guide

CDI and Dependency Injection

Score: 77



Description:

Covers the use of Contexts and Dependency Injection to manage the creation, lifecycle, and wiring of application components. Includes defining and injecting beans, working with scopes, using qualifiers, and leveraging CDI events and interceptors.

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 CDI and Dependency Injection in Jakarta EE, including bean definition and injection, scope management, and the use of qualifiers. Some proficiency with more advanced topics such as CDI events and interceptors is evident, though mastery of all areas may not be fully consistent.

Explain how you would choose the right CDI bean scope for a component that holds user-specific data during a web session, and describe what could go wrong if you chose an incorrect scope.



1

Cannot name CDI scopes or distinguish between @RequestScoped, @SessionScoped, and @ApplicationScope.



2

Identifies @SessionScoped as correct but cannot articulate consequences of misusing @ApplicationScoped for user data.



3



4

Clearly compares scopes, identifies shared-state bugs from @ApplicationScope, and mentions passivation requirements for @SessionScoped.



5

What is dependency injection, and how would you use CDI to inject one bean into another in a Jakarta EE application?



1

Cannot explain dependency injection or identify the @Inject annotation.



2

Correctly uses @Inject but cannot explain bean scopes or why CDI is preferable to manual instantiation.



3



4

Explains IoC principle, uses @Inject with appropriate scope annotations, and describes benefits for testability and loose coupling.



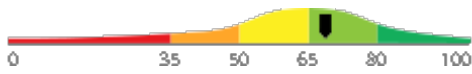
5

Detail

Interview Guide

JAX-RS and RESTful API Development

Score: 69



Description:

Covers the design and implementation of RESTful web services using the JAX-RS API. Includes defining resource classes and methods, working with HTTP verbs, path parameters, query parameters, and producing or consuming JSON and XML payloads.

Interpretation:

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

The candidate exhibits a solid and well-rounded understanding of JAX-RS and RESTful web service implementation within the Jakarta EE framework. They are likely proficient in defining resource classes and methods, working with HTTP verbs, managing path and query parameters, and handling JSON and XML payloads effectively.

Walk me through how you would design and implement a RESTful endpoint that accepts a JSON request body, validates the input, and returns appropriate HTTP status codes for both success and error scenarios.



1

Cannot map JSON to a Java object or select correct HTTP status codes; no mention of validation.



2

Uses @POST and @Consumes correctly and returns basic status codes but lacks input validation or error response detail.



3



4

Combines @POST, @Consumes, Bean Validation, and Response builder with meaningful status codes and error payloads.



5

What is JAX-RS, and how would you create a simple REST endpoint in a Jakarta EE application that returns a list of items in JSON format?



1

Cannot identify JAX-RS or confuses it with other frameworks; no knowledge of annotations.



2

Identifies JAX-RS and key annotations like @GET and @Path but struggles with JSON serialization or application config.



3



4

Correctly uses @Path, @GET, @Produces, and explains Application subclass or web.xml configuration with JSON binding.

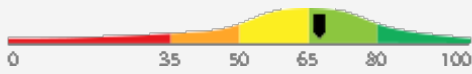


5

Detail Interview Guide

JPA and Database Persistence

Score: 67



Description:

Covers the use of the Java Persistence API to map Java objects to relational database tables and manage their lifecycle. Includes defining entities, configuring relationships, writing JPQL queries, and using the EntityManager to perform CRUD operations.

Interpretation:

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

The candidate demonstrates a solid and proficient understanding of JPA and relational database persistence principles. They are likely capable of defining and mapping entities, configuring relationships, writing JPQL queries, and managing entity lifecycles using the EntityManager with only occasional gaps in more advanced or nuanced areas.

Describe how you would model and query a one-to-many relationship between two entities in JPA, and explain how you would avoid common performance problems such as the N+1 query issue.



1

Cannot map a one-to-many relationship or is unaware of the N+1 problem.



2

Maps the relationship with @OneToMany but cannot explain fetch strategies or JOIN FETCH to solve N+1.



3



4

Correctly maps bidirectional relationship, explains LAZY vs EAGER fetching, and uses JOIN FETCH or EntityGraph to optimize queries.



5

What is an entity in JPA, and how would you map a simple Java class to a database table so that you can save and retrieve records?



1

Cannot define an entity or identify required annotations like @Entity and @Id.



2

Correctly annotates an entity with @Entity and @Id but cannot explain persistence context or basic CRUD with EntityManager.



3



4

Defines entity mapping, explains @GeneratedValue, persistence context, and demonstrates EntityManager persist/find/remove.



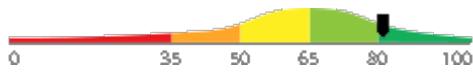
5

Detail

Interview Guide

Servlets, JSP, and HTTP Request/Response Handling

Score: 81



Description:

Covers the creation and configuration of servlets and JSP pages to handle HTTP requests and responses. Includes session management, request dispatching, filters, and listeners. These are the core building blocks of any Jakarta EE web application.

Interpretation:

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

The candidate exhibits an advanced and comprehensive mastery of Java Jakarta EE programming, demonstrating strong proficiency across the full spectrum of assessed competencies, including RESTful API design, CDI frameworks, JPA entity management, security mechanisms, and application server configuration. They are well-equipped to independently design, develop, debug, and maintain complex business applications within the Jakarta EE ecosystem, applying sound judgment in areas such as transaction management, unit testing, and data interchange. This individual is expected to perform at or above a mid-level standard and may be capable of mentoring others or leading technical efforts on Jakarta EE-based projects.

Describe a situation where you needed to share data across multiple requests for the same user in a Jakarta EE web application. How did you implement session management, and what pitfalls did you watch out for?



1

Vague answer; cannot distinguish session from request scope or identify common session pitfalls.



2

Correctly uses HttpSession but misses concurrency issues, session expiry, or security concerns.



3



4

Demonstrates confident use of HttpSession, addresses thread safety, session fixation, and invalidation strategies.



5

Can you explain what a servlet is and describe the basic steps you would take to handle an HTTP GET request in a Jakarta EE application?



1

Cannot define a servlet or confuses GET/POST; no understanding of lifecycle.



2

Defines a servlet and outlines basic GET handling but misses lifecycle or configuration details.



3



4

Clearly explains servlet lifecycle, annotations like @WebServlet, and correctly handles GET with HttpServletRequest/Response.



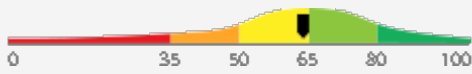
5

Detail

Interview Guide

Transaction Management and Data Access Patterns

Score: 64



Description:

Covers how Jakarta EE manages database transactions to keep data consistent and reliable. Includes using container-managed transactions with EJB or CDI, applying transactional boundaries, handling rollbacks, and implementing common data access patterns such as the Repository or DAO pattern.

Interpretation:

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

The candidate possesses a basic to moderate understanding of transaction management and data access patterns in Jakarta EE. They are likely familiar with core concepts such as container-managed transactions and common data access patterns, though consistent and accurate application of these concepts may still require development.

Describe how you would structure data access code in a mid-sized Jakarta EE application to keep business logic separate from database operations, and explain how transaction boundaries fit into that structure.



1

Places SQL or EntityManager calls directly in servlets or REST resources with no separation of concerns.



2

Uses a DAO or Repository layer but cannot clearly explain where @Transactional should be applied or why.



3



4

Implements a clean service/repository layering, applies @Transactional at the service layer, and explains propagation across nested calls.



5

What is a database transaction, and how would you ensure that a sequence of database operations either all succeed or all roll back together in a Jakarta EE application?



1

Cannot define a transaction or identify any Jakarta EE mechanism for managing one.



2

Understands the concept of atomicity and mentions @Transactional but cannot explain rollback rules or propagation.



3



4

Explains ACID properties, correctly applies @Transactional, describes rollbackOn settings, and explains default propagation behavior.



5

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
<p>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.</p> <p>The program declares a function called <code>duplicate_array</code> that:</p> <ol style="list-style-type: none">1. Takes a const int pointer to a source array and its length as parameters.2. Uses <code>calloc</code> to allocate a new int array of the same length.3. Returns NULL if <code>calloc</code> 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. <p>In main, the program:</p> <ol style="list-style-type: none">1. Declares and initializes a stack array of 4 integers with values 5, 15, 25, 35.2. Calls <code>duplicate_array</code> 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. <p>Use appropriate indentation, common C coding conventions, and add brief inline comments where needed. Type your completed source code as your response.</p>	<pre>#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; }</pre>

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

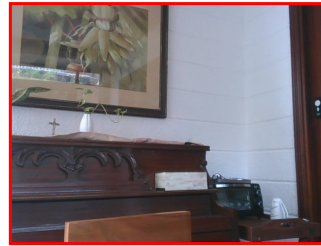
- Risk:	Medium risk of cheating based on image inconsistencies
- Percent match among processed faces	100%
- Total images processed	17
- Total images with valid faces	14 (82%)
- Total pairs of faces compared	13
- Pairs in which faces matched	13 (100%)



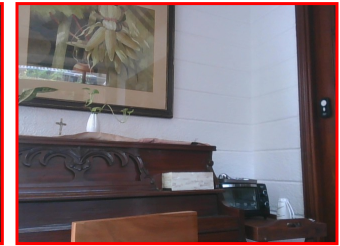
Pre/Post-Test Photo



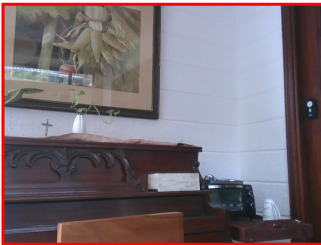
ID Photo



In-Test Error Detected (No Face Detected)



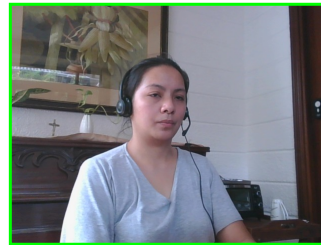
In-Test Error Detected (No Face Detected)



In-Test Error Detected (No Face Detected)



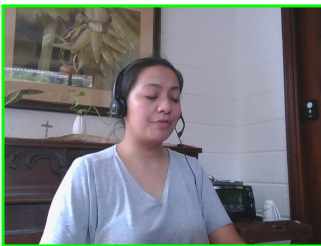
In-Test Photo



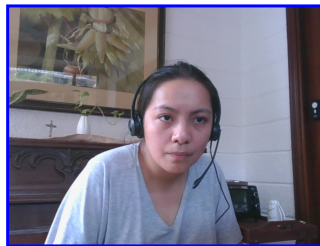
In-Test Photo



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: 20664-1, Key: 0-0, Rpt: 68, Prd: 9542, Created: 2026-06-25 14:24 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 (%)
Bean Validation and Exception Handling	64.8866	Numeric Score	64.8866	12.5000
CDI and Dependency Injection	77.0141	Numeric Score	77.0141	12.5000
JAX-RS and RESTful API Development	69.0008	Numeric Score	69.0008	12.5000
JAX-RS and RESTful API Development (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
JPA and Database Persistence	67.9734	Numeric Score	67.9734	12.5000
Servlets, JSP, and HTTP Request/Response Handling	81.2743	Numeric Score	81.2743	12.5000
Servlets, JSP, and HTTP Request/Response Handling (Coding Tasks)	62.9784	Numeric Score	62.9784	12.5000
Transaction Management and Data Access Patterns	64.1854	Numeric Score	64.1854	12.5000
Weighted Average:				68.7864
Final Overall Score:				68

Notes

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