

Test Results and Interview Guide


Candidate: **Elizabeth Wantsajob**
Assessment: Object Oriented Programming Concepts (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 Object Oriented Programming Concepts (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 Object Oriented Programming Concepts (Short) June 28, 2026	<div style="background-color: #28a745; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">82</div>	





The candidate exhibits an advanced and comprehensive mastery of object-oriented programming concepts, including sophisticated topics such as complex design patterns, UML diagram interpretation, generics, coupling and cohesion, and fine-grained access control. They are well-equipped to independently architect, develop, and maintain robust object-oriented systems and may be positioned to serve as a technical resource or mentor for other team members.

Key

- Higher Risk
- Lower Risk





▼ Candidate Score

Competency Summary

Competency	Score	Interpretation
Skills/Knowledge (relates to immediate readiness)		
Abstraction and Interfaces	73	
Classes and Objects	94	
Encapsulation and Access Modifiers	73	
Inheritance and Polymorphism	89	

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	82nd												
North America	68th												
United States	68th												
Example Company	76th												

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: Object Oriented Programming Concepts (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: 82

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

Abstraction and Interfaces

Score: 73



Description:

Understanding how to define abstract classes and interfaces to establish shared contracts between components. Includes knowing when to use an interface versus an abstract class and how these tools help separate what a component does from how it does it.

Interpretation:

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

The candidate exhibits a solid and proficient understanding of abstraction and interfaces, including when to use each construct and how they help separate a component's contract from its implementation. Minor gaps in knowledge or application may exist, but they are capable of working effectively with these concepts in most professional contexts.

How would you decide whether to use an abstract class or an interface when designing a set of related components? Walk me through your reasoning.

- ★
1
- ★
2
- ★
3
- ★
4
- ★
5

Cannot articulate a meaningful difference or applies the wrong tool to a described scenario.

Identifies key differences and makes a reasonable choice but reasoning is incomplete or partially incorrect.

Clearly articulates the tradeoffs, references shared state or default behavior vs. pure contracts, and applies sound reasoning.

What is an interface, and how is it different from a regular class? Why might you use one?

- ★
1
- ★
2
- ★
3
- ★
4
- ★
5

Cannot define an interface or confuses it with a class or abstract class.

Correctly defines an interface and notes it cannot be instantiated but gives a limited reason for using one.

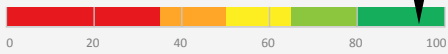
Clearly defines an interface, distinguishes it from a class, and explains its role in defining contracts between components.

Detail

Interview Guide

Classes and Objects

Score: 94

*Description:*

Understanding how to define classes and create objects from them. Includes knowledge of constructors, instance variables, methods, and how objects are instantiated and used in a program.

Interpretation:

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

The candidate exhibits a comprehensive and advanced understanding of object-oriented programming concepts, including class design, design patterns, SOLID principles, exception handling hierarchies, object lifecycle management, and UML interpretation. They demonstrate the ability to confidently design, develop, and maintain robust, maintainable, and well-structured object-oriented programs.

Walk me through how you would design a class for a real-world entity you have worked with. What fields, constructors, and methods would you include and why?



1

Describes a vague or incomplete class design with little reasoning behind design choices.



2

Describes a reasonable class design with some justification for fields, constructors, and methods.



3



4

Provides a well-structured class design with clear reasoning, appropriate use of constructors, and encapsulated fields.



5

Can you explain what a class is and how it differs from an object? Can you give a simple real-world example of each?



1

Confuses classes and objects or cannot provide a meaningful distinction or example.



2

Correctly distinguishes classes from objects but explanation or example lacks clarity.



3



4

Clearly explains class as a blueprint and object as an instance, with a strong, concrete example.



5

Detail
Interview Guide
Encapsulation and Access Modifiers

Score: 73


Description:

Understanding how to protect the internal state of a class by restricting direct access to its fields and methods. Includes knowing how to use access modifiers such as public, private, and protected, and how to expose data safely through getters and setters.

Interpretation:

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

The candidate demonstrates a solid understanding of encapsulation principles and the use of access modifiers, including public, private, and protected. They are generally proficient in designing classes that protect internal state and expose data safely through getters and setters, with only minor gaps in knowledge.

Describe a situation where poor encapsulation caused or could cause a bug or maintenance problem in a codebase. How would you fix it?



1

Cannot describe a realistic scenario or fix, or conflates encapsulation with unrelated concepts.



2

Describes a plausible scenario with a reasonable fix but lacks depth in the explanation.



3



4

Gives a specific, realistic scenario and a clear fix using proper access modifiers and controlled access methods.



5

What does it mean to make a field private in a class, and why would you do that instead of making it public?



1

Cannot explain the difference or gives an incorrect reason for using private fields.



2

Correctly identifies that private restricts access but gives a limited explanation of why it matters.



3



4

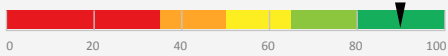
Clearly explains data hiding, preventing unintended modification, and the role of getters and setters.



5

Inheritance and Polymorphism

Score: 89


Description:

Understanding how classes can inherit fields and methods from parent classes, and how objects of different types can be treated as instances of a common parent type. Includes knowledge of method overriding and how polymorphism allows flexible, reusable code.

Interpretation:

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

The candidate demonstrates a comprehensive and advanced understanding of inheritance and polymorphism within object-oriented programming. They are well-versed in class hierarchies, method overriding, and leveraging polymorphism to write flexible, reusable, and maintainable code. This level of proficiency reflects strong readiness to apply and potentially mentor others in these concepts within a professional environment.

Can you explain the difference between method overloading and method overriding? When would you use each one?



1

Confuses the two concepts or cannot explain when each would be used.



2

Correctly distinguishes the two but gives limited or generic examples of when to use them.



3



4

Clearly explains both concepts with accurate examples and articulates appropriate use cases for each.



5

What is inheritance in object-oriented programming, and can you give an example of when you would use it?



1

Cannot define inheritance or provides an inaccurate or nonsensical example.



2

Correctly defines inheritance and provides a basic example but does not explain the practical benefit.



3



4

Clearly defines inheritance, gives a relevant example, and explains how it promotes code reuse.



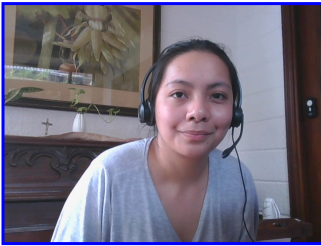
5

Identity Confirmation Photos

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

Photo Analysis Results

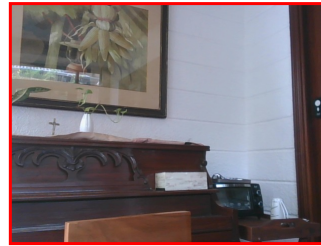
- Risk:	Medium risk of cheating based on image inconsistencies
- Percent match among processed faces	100%
- Total images processed	17
- Total images with valid faces	14 (82%)
- Total pairs of faces compared	13
- Pairs in which faces matched	13 (100%)



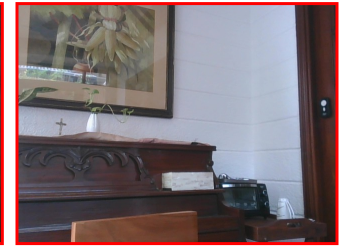
Pre/Post-Test Photo



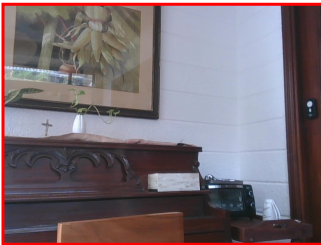
ID Photo



In-Test Error Detected (No Face Detected)



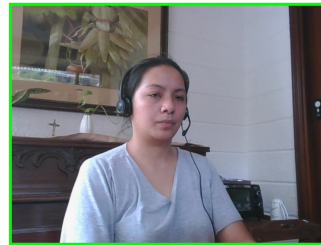
In-Test Error Detected (No Face Detected)



In-Test Error Detected (No Face Detected)



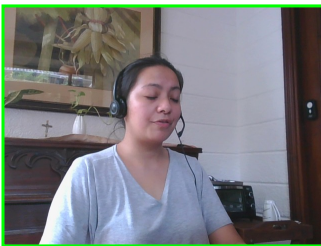
In-Test Photo



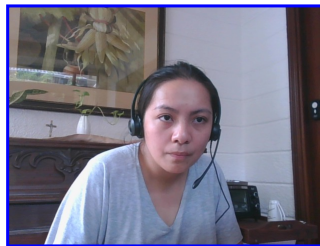
In-Test Photo



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: 20814-1, Key: 0-0, Rpt: 68, Prd: 9636, 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 (%)
Abstraction and Interfaces	73.9116	Numeric Score	73.9116	25.0000
Classes and Objects	94.0727	Numeric Score	94.0727	25.0000
Encapsulation and Access Modifiers	73.6447	Numeric Score	73.6447	25.0000
Inheritance and Polymorphism	89.6709	Numeric Score	89.6709	25.0000
Weighted Average:				82.8250
Final Overall Score:				82

Notes

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