

Byteboard SWE Interview Prep Guide

Before you start

- Read the [FAQ](#)
- Choose a coding language you are comfortable coding in (Java, C++, Python, Ruby, Go, PHP, C#, JavaScript)
- Inform your recruiter about any accommodation requests

The Byteboard interview includes two parts:

Part 1: Technical Reasoning Exercise

You will be working on a document that describes a new product your team is planning to launch. The document outlines several different implementation options. You will **complete the 'Conclusion' section and discuss which option(s) your team should pursue** with reasoning behind your decision. The document also contains **comments you will answer** to help your team understand the motivations behind various decisions.

Read	Respond	Conclude
Spend ~15 mins	Spend ~10 mins	Spend ~15 mins
Add comments with questions or clarifications along the way	Carefully respond to the comments left in the doc	No right or wrong answer
	Fully state your assumptions	Pick an option
		Clearly communicate why you chose the option over pros/cons of other options

Part 2: Code Implementation Exercise

You will **implement a few aspects of the project** featured in part one. You will need to **navigate a small code base** and **write clean and functional code**.

Read	Identify	Implement
<p>Read through the codebase, build a mental model of how the code fits together</p> <p>Recommended take at least ~20 mins</p>	<p>Identify what algorithm / data structure you are going to use</p>	<p>What's the simplest way to complete the task?</p>

Helpful Tips

For Part 1:

- Your interview will be evaluated by a real engineer. To ensure they understand your thought process:
 - Fully state your assumptions.
 - Add comments in the document for any clarifications or questions you have about the design.
 - Do not spend too much time responding to any one question.
 - Complete the design document to the best of your ability and come back to expand on answers if you have more time.
- If there's a concept you are not familiar with, Google it!

For Part 2:

- Spend some time reading through the codebase and building a mental model of how the code fits together.
- Run your code often to detect bugs early on.
 - Pay attention to edge cases.
 - Try to submit code that does not crash.
 - Consider adding test cases to catch edge cases or a future regression.
- Your interview will be evaluated by a real engineer meaning all of your progress (including comments!) will be taken into account. To ensure they understand your thought process:
 - Always state your assumptions, questions, or thoughts in clear comments.
 - If you do not finish all the tasks or have additional ideas for optimizations, describe what you would do with more time by adding comments.
- If you forget syntax or encounter a concept you are not familiar with, Google it!

How to Prepare

- Look through past projects, work or school assignments you've done. Why did you approach the problem the way you did?
- Understand the difference between basic data structures and when it's beneficial to use each.
- Take your time to read through the material for each part and tasks.
- Pick the language for which you feel the most comfort reading and writing code in a multi-file project. Review syntax if it's been a while!
- Get comfortable working in your chosen language in your favorite IDE. You will also have the option to use a web based IDE if you prefer
- Overcommunicate! Tell us what you are thinking throughout the interview. Ask questions / add comments in Part 1 of the interview and write comments in your code telling us what you are thinking in Part 2.