Interviewing at Bloomberg

Our engineers and recruiters created this document to help you understand what to expect and how best to prepare. Remember, it's as much about you assessing us as it is about us assessing you. We're looking forward to our next conversation because we think you'll be an asset to the business!

Phone tech screen:

The tech screen is a one hour assessment designed to assess your communication and technical skills. We understand that your coding ability can differ from that of a real work setting vs solving new problems under time constraints so, it's important to practice and familiarise yourself with assessment platforms. This guide will highlight hints and tips to help you put your best foot forward.

Virtual onsite:

The onsite focuses on multiple aspects, however the first 2 sessions will typically be technically focused on software engineering fundamentals. Session one will typically be coding and session two would typically be high and low level systems design (whiteboard). We'll let you know in advance if your schedule is different. After the technical assessments, you'll then have a break. During this time, the team will reconvene to assess technical performance and decide if they'd like to move to the second half of the interview.

Additional rounds will usually be conducted with Leads, Managers, and Group/Org Managers to assess leadership, motivations, career goals, communication, design and architecture skills. You can expect questions related to your work experience, solutions you developed, and your approach to simulated scenarios. Your recruiter will confirm the schedule during a prep call.

Post-Onsite:

The hiring team will typically debrief within 24-48 hours after your interview (depending on interviewer availability). The team will come to a decision and you should expect to hear from your Recruiter within a week after the interview.

Bloomberg Engineering

TechAtBloomberg.com

Technical Assessment - Coding

Coding

- We ask questions with multiple possible solutions to evaluate coding fluency, problem-solving, common data structures and use of common algorithms
- We care more about how you get there and what your methods are than perfect syntax or optimal solutions on your first try. Most teams won't run the code but clean coding style is still important.
- You can use the programming language you are most familiar with and you can expect some language specific questions to assess the depth of your knowledge.
- Most teams will aim to complete one if not two challenges.

Tips

- Before diving into coding, always discuss the question with your interviewer. Ask clarifying questions to make sure you understand the entire problem.
- Make sure to consider edge cases in your solutions - you won't have to handle them all, but discuss them with your interviewer.
- Don't worry about your solution being perfect because time is limited. Write what comes but then refine it later.
- If you have time, think about refactoring/optimising your code and writing it in a way that would be easier to unit test or communicate with the interviewer on what you'd do if there was more time.
- We use Hackerrank as our testing platform but leetcode is a great platform to research and take technical challenges.

Bloomberg Engineering

How to prepare - Coding

During this session we may cover a number of topics such as complexity, data structures, object oriented features (polymorphism, encapsulation, inheritance), memory management, error handling, multithreading and much more.

These exercises assume you have knowledge in coding but not necessarily knowledge of data structures and algorithms.

Coding Practice -

- Arrays & Strings
  - ➢ A very big sum - Entry level - Familiarise yourself with the platform
  - ➢ Let Rotation

- Lists - Learn the basics here
  - ➢ Insert a Node at a Position Given in a List
  - ➢ Cycle Detection

- Stacks & Queues - Learn the basics here
  - ➢ Balanced Brackets
  - ➢ Queue Using Two Stacks

- Hash & Maps - Learn the basics here
  - ➢ Ice Cream Parlour

- Sorting Algorithms - Learn the basics here
  - ➢ Insertion Sort - Part 1
  - ➢ Insertion Sort - Part 2

- Trees - Learn the basics here
  - ➢ Binary Tree Insertion
  - ➢ Height of a Binary Tree

- Graphs (BFS & DFS) - Learn about Graphs here
- Breadth First Search
- Snakes and Ladders

- Recursion - Review concepts on recursion here
- Fibonacci Numbers

TechAtBloomberg.com

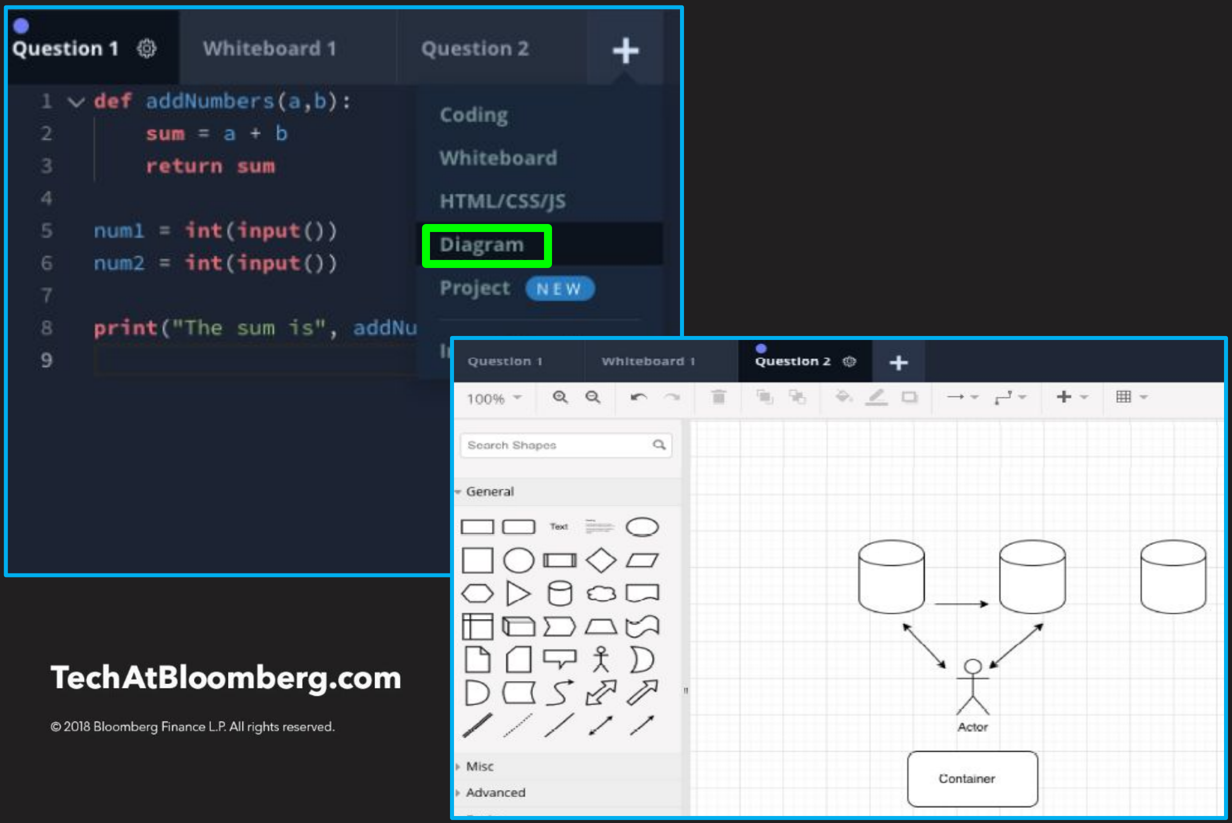
Technical Assessment - Design

System Design

- System design questions are used to assess a candidate's ability to combine knowledge, theory, experience and judgement towards solving a real-world engineering problem. In other words, the goal is to understand whether a candidate can design policies, processes, procedures, methods, tests, and/or components from ground up.

Tips

- You will be presented with open-ended questions, so always remember to **ask clarifying questions** and solidify you know the parameters of the problem before you jump into solving it.
- As with the coding rounds - Keep things simple!
- You can refer to the following 4 step framework when considering your approach to a design question:
    1. **Problem** - what is the business problem you're trying to solve?
    2. **Design** - how will you design a system to solve that problem?
    3. **Implementation** - which tools/technologies will you use? What are the low-level CS problems around the system?
    4. **Optimisation** - how will you make it better/faster?
- HackerRank has a diagram component (see screenshot below) where you can leverage "smartart" shapes to describe how you would design a system (or how you've designed systems in the past). The most important thing to remember is to talk through your solutions and explain your thought process.



```
def addNumbers(a,b):
    sum = a + b
    return sum

num1 = int(input())
num2 = int(input())

print("The sum is", addNu
```

Bloomberg Engineering

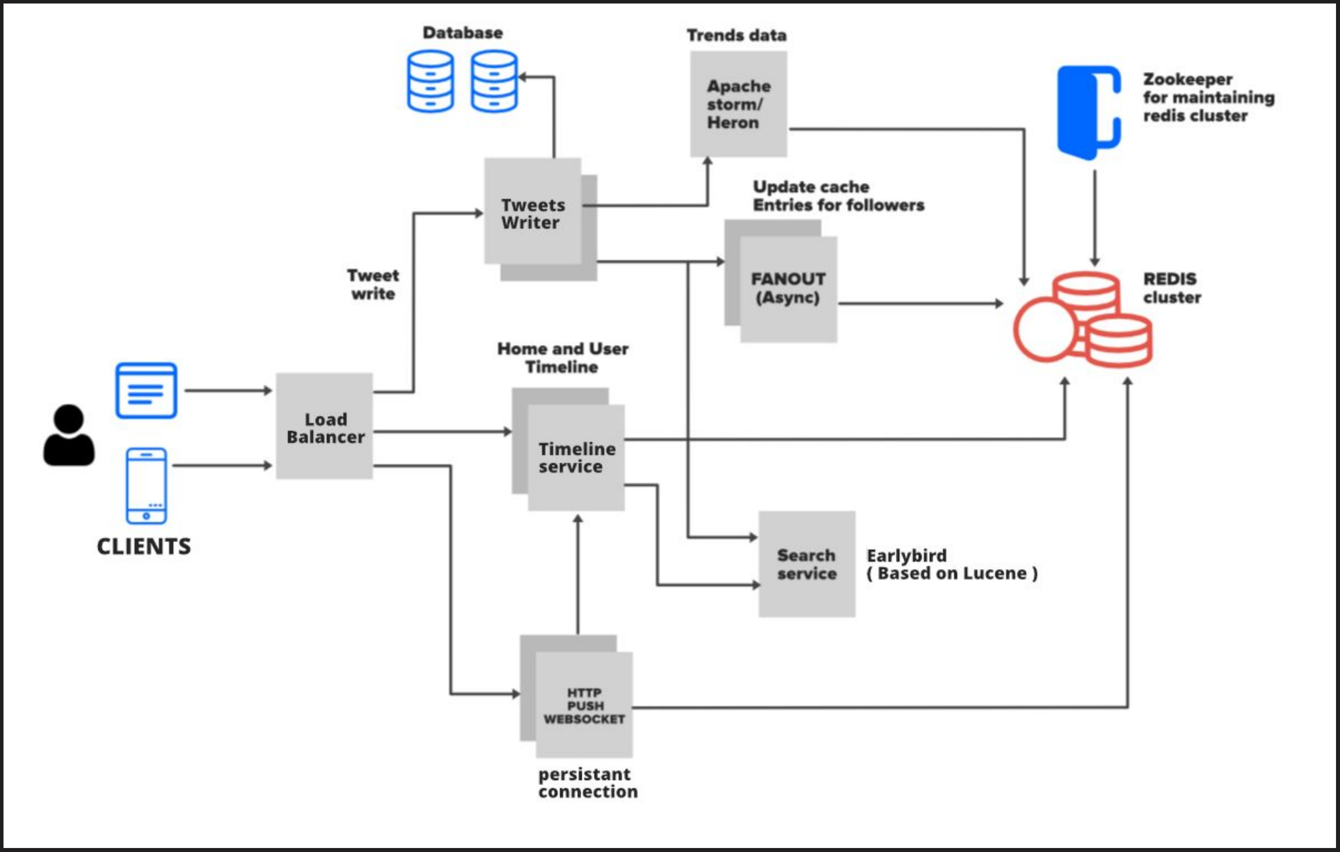
How to prepare - Design

In this session we'll assess your ability to go from high-level requirements to an actionable design solution and then low level problem solving.

- Try and drive the discussion but be open to suggestions
- Break down the problem into manageable chunks
- Gather requirements and ask clarifying questions
- Explain and reason your decisions
- Adapt the architecture to changed requirements
- Clearly communicate your ideas
- Always think scalability, reliability and testing strategies

Design Practice - Inhouse

- How would you design a social media platform like Twitter?



TechAtBloomberg.com

Bloomberg

General Interview Tips

- Keep a pen & paper or computer nearby to take notes or organize your thoughts.
- Don’t be afraid to ask your interviewer for help or a nudge in the right direction - they are there to help!
- If the interviewer gives you hints to improve your code, take them and run with them! It’s good to adjust and work through the problems with the interviewer to show your collaboration skills.
- Take a second to collect your thoughts before answering. It’s ok to ask the interviewer for a bit of time to organize yourself before you speak.
- Try to think out loud if you are mentally working through a solution and always be sure to explain your thought process when solving a problem. While getting the correct answer is important, we want to understand your methodology.
- Use the STAR (Situation, Task, Action, Result) technique when answering example based questions
- If you have any questions regarding our company working environment, expectations, usage of specific technologies, or any other curiosities regarding our work here, please do take your chance to ask the interviewers. We also suggest asking the same questions per session as each interviewer will probably offer you something different.
- Remember: every interviewer you meet with had to go through a similar process to join Bloomberg!

Motivations

- Be prepared to talk about your motivations for wanting to join Bloomberg, or equally reasons for why you want to leave your current role.
- If you were not active in your search for a new job and a Bloomberg employee reached out to you directly, it’s still important to be able to give clear reasons and motivations for interviewing.
- It's also a good idea to be able to talk about what you hope to work on in your next role, talk about personal areas of interest, or specific projects you enjoyed in previous roles.

Bloomberg Engineering

More Resources

Books

- Cracking the Coding Interview
  Gayle Laakmann McDowell
- Programming Interviews Exposed: Secrets to Landing Your Next Job
  John Mongan, Eric Giguere, Noah Suojanen, Noah Kindler
- Programming Pearls
  Jon Bentley
- Introduction to Algorithms
  Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein

Practice

- Geeks for Geeks: http://www.geeksforgeeks.org/
- Career cup: http://www.careercup.com/page
- Code Chef: http://www.codechef.com/
- Project Euler: https://projecteuler.net/
- Leetcode: https://leetcode.com/problemset/algorithms/
- More algorithmic challenges! http://www.algorithmist.com/

About Bloomberg

- What we do - https://www.bloomberg.com/company/what-we-do/
- Tech at Bloomberg - https://www.techatbloomberg.com/blog/
- Values - Philanthropy, Innovation, D&I and Sustainability
- Testing - https://www.hackerrank.com/test/sample
- Twitter - https://twitter.com/TechAtBloomberg

Bloomberg Engineering

TechAtBloomberg.com