FLATIRON

Flatiron Health, Inc.
200 5th Ave.
New York, NY 10010

# Preparing for your Flatiron Software Engineering Interview

Thank you for taking the time to interview with Flatiron Health. Our engineers are tackling a wide range of challenging engineering problems to organize the world's oncology information and make it useful for doctors, patients, researchers and life science companies. We're hoping your technical interview will be as fun as it is challenging. This document is designed to give you a sense of what to expect and to help you to represent your skills in the best light. It applies to technical screens (which consist of a coding interview) as well as on-site interviews (which consist of multiple coding and other interviews).

If you have not been interviewing recently, we recommend practicing some tech interview problems on your own. There are a number of online resources that offer sample algorithmic and OO design problems such as www.codechef.com and www.topcoder.com.

## Coding interviews (screen and on-site):

**Format**
You will be posed a computer science problem and asked to design and implement an efficient solution.

During a screen, you will be given a link to a paired session with your interviewer to write your code.

During on-site interviews, coding will be done on a whiteboard. We recommend practicing solving coding problems with pen and paper to simulate not having a computer.

**Programming Languages**
We do not require knowledge of any specific programming language; our interview process is language-agnostic. However, you should be strong in one object-oriented language such as Java, Python, C#, C++, or Ruby.

Make sure you choose the programming language that you are most comfortable with. We will be looking to see if you can write syntactically correct code, find your own bugs and check for edge cases.

**Data Structures**
You should be familiar with common data structures and be able to discuss the tradeoffs between them.

**Algorithms**
You should be familiar with common algorithms (e.g. sorting, binary search, BFS/DFS) and the situations in which they are applicable. You will be expected to characterize the runtime complexity of code that you write during your interview. Don't worry about memorizing the specific details for any particular algorithm - we won't ask you to implement a red-black tree or simulated annealing. We're looking to see that you understand the tradeoffs and limitations of the code you write and the third-party libraries you use.

## Additional technical interviews (on-site only):

In addition to the whiteboard coding interview, you may have one or two of the following interview types, depending on your specific skill set and experience level. Talk to your recruiter about which of these interviews to expect.

**Object-Oriented Design**
We want to see if you understand OO best practices and we will assess your knowledge of common design patterns. Be prepared to discuss tradeoffs and defend your design.

**Systems Design**
This interview tests your knowledge of basic distributed computing concepts, such as service-oriented architectures, distributed caching, load balancing, etc. You also should be familiar with some OS topics that can affect code performance, such as memory management, processes, threads, synchronization, paging and multithreading.

**Web Development**
This interview tests your knowledge of web application architecture, JavaScript, asynchronous programming and UI development patterns.