Per our conversation, the phone interview will take an hour and you will need to have a computer with internet access for the coding portion of the interview, done through www.collabedit.com.

Major components of our assessment involve:
- Applying algorithm, data structures and Computer Science fundamentals
- Explanation and implementation of the solution in code without any of the aids engineers typically have access to (e.g. IDE, online docs).
- Pace-- you will be under the time gun.

Algorithms hints:
- Think about the trade-offs of different approaches.
- Knowing the run-time complexity of your solution may indicate whether you can do better.
- Don't over-complicate your algorithm to fit in specific concepts - we are looking for answers to the question, not looking for certain techniques.

Coding sites to visit for practice questions:
1. Leet Code
2. Hacking the Google Interview
3. Top Coder
4. Developers book (Core Java)
5. Stack Overflow

I know interviewing can be a stressful process, so here are a few additional tips to help you:
Tip #1: ***Ask*** clarifying questions instead of making assumptions. Before starting on your implementation, be clear on requirements. Describe your high level approach first before implementing.

Tip #2: ***Verbalize*** your thoughts out loud while you're solving a problem. Our interviewer wants to understand your thought process and how you go about solving a problem. So by thinking out loud, you'll give them an idea and they'll be able to give you hints to steer you in the right direction. Clear technical communication is key. Give your interviewer an idea of what they should expect before jumping into the solution. Take notice of suggested hints.

Tip #3: You will be assessed on both ***speed*** as well as ***quality*** of the solution. So, don't take too much time to think through the problem, but take enough time so you can provide a clean and elegant solution. For some questions, the expectation is that someone should be able to work through the problem in "X" minutes, so there could be multiple questions in the hour slot; others may take the entire hour to solve.

Tip #4: If asked about an interesting project, make sure to choose a project that you are actually interested in rather than just picking your most current project. Show ***enthusiasm*** *and* ***details*** about the project by explaining who was your customer, why the project/product was being built, and give detailed information about your contributions and specific responsibilities.

Tip #5: ***Find and fix*** your bugs using edge cases, tests, or other means.

Tip #6: ***Visit*** our engineering blog to find out a bit more about the different technologies and projects we are working on as well as get a feel for our engineering and company culture. Here are some helpful links to learning more about LinkedIn Engineering:

Engineering Blog
Data @ LinkedIn
Open Source @ LinkedIn
LinkedIn's Culture
Engineering Insights (Video interviews with engineers at LinkedIn)

Tip #7: Review your ***data structures*** and the libraries for them in your language(s) of choice, particularly paying attention to their limitations.

Tip #8: Ask relevant and appropriate questions about LinkedIn to show your interest. This is your chance to ***interview us*** to make sure that LinkedIn is the right fit for you.

Lastly, below are links to posts written by our engineers. They give a lot of info and provide different perspectives on how to prepare for this type of interview:
What to do (and not do) if I interview you
Interviews are not Exams
Getting That Job At LinkedIn (I strongly recommend that you spend the most time reviewing this article.)
Technical Design Interview
Success in the Coding Interview