Preparing for your Technical Phone-Screen.

Our Phone-Screen usually involves 1 (sometimes 2) coding questions where you will be asked to write a solution.

A few very important factors as part of this screening that you can do as a candidate during your coding question include the following: The solution will be to test your understanding of Data Structures and Algorithms – these will be main focus areas. The 5 highlighted bullet-points below outline some of the things that you will be evaluated upon during the phone-screen, so these can likely make a big difference in the overall success of your interview! (Please Review)

1. **Clarify the initial requirements/question** (repeat exactly what the interviewer is looking for and talk out loud what requirements you would like clarification on so you know how to best approach the problem. (Do NOT jump right into coding the solution – this happens and sometimes it does not go very well.) We at Amazon are always looking at Solutions that are not over complicated – Invent and Simplify is one of our critical leadership Principles.
2. **Ask follow-up questions** while you are designing your solution (In fact interviewers really like when you ask follow-up questions.)
3. **Be sure to take direction / hint's well from the interviewer** – they ultimately want to see you succeed (This is something that we evaluate as a team member – we like to hire software engineer's that are able to take feedback / direction and follow it when necessary. This shows that you are a good team-member and would be someone they could work with.
4. **When solving the problem these are good things to think / talk out loud** about with the interviewer. (Things that are great to look at regarding your solution would be (**Is it Scalable, Look for Bugs/Talk about importance of Test Cases.)**
5. **Finally solve the problem** within a reasonable amount of time. At the end you could perhaps discuss ways that you could have done things better for **more extensibility / optimization.**

- **Architecture and Software and Service Design**

Understanding of Service Oriented Architecture and experience of a service that your team has built. Characteristics of a good API. Scalability. Integration with existing platforms.

- **Programming Languages**

We do not require that you know any specific language before interviewing for a technical position at Amazon.com, but familiarity with a prominent object oriented language is generally a prerequisite for success. Not only should you be familiar with the syntax of a language like C++, Java, or C#, you should also know some of the language nuances such as how memory management works, what some of the most commonly used collections or libraries are, etc. You should be able to compare languages and talk about the tradeoffs between using language X vs. language Y.

Additionally, it's considered a plus to be familiar with some scripting language such as perl, ruby, etc. It's also nice to know the basics of regular expression as they are now a mainstay in both the object oriented and scripting worlds.

- **Data Structures**

Most of the work we do involves storing and providing access to data in efficient ways. **This necessitates a very strong background in standard data structures**. You should know what each of these data structures is and how they're implemented; what their runtimes are for common operations; and under what circumstances it would be beneficial to use one. The below are in no particular order.

- **Array**
- **Linked List**
- **Tree (Tree, Binary Tree, Binary Search Tree, Red-Black Tree, etc.)**
- **Heap**
- **Hash Table**
- **Stack**
- **Queue**
- **Tree**
- **Graph (both directed and undirected)**

- **Algorithms**
  - It's also important to know efficient ways manipulate data. One great way of doing this is brushing up on some common algorithms. We'll expect that you can apply and discuss the tradeoffs between some commonly used algorithms.
- **Sorting**
  - Bubble Sort
  - Merge Sort
  - Quick Sort
  - Radix/Bucket Sort

- **Traversals (On multiple data structures)**
  - Depth First Search
  - Breadth First Search

- **Coding**

Expect to be asked to code syntactically correct code – no pseudo code. If you're a bit rusty coding without an IDE or coding in a specific language, it's probably a good idea to dust off the cobwebs and get comfortable coding with pen and paper. The most important thing a software engineer does at Amazon.com is write scalable, stable, robust, and well tested code. These are going to be the main criteria by which your code will be evaluated, so make sure that you check for edge cases and common error inputs as well as the "happy paths" through the code.

- **Object Oriented Design**

Good design is paramount to extensible, bug free, and long living code. It's possible to solve a software problem in an almost limitless number of ways, but when software needs to be robust and extensible, it's important to know some common techniques that help with this. Using object oriented design best

practices is one way to build lasting software. You should have a working knowledge of a few common and useful design patterns (singleton, factory, adapter, bridge, visitor, command, proxy, observer, etc.) as well as know how to write software in an object oriented way with appropriate use of inheritance and aggregation.

- **Databases**

Most of the software that we write is backed by a database somewhere. A lot of the challenges we face come in to play when interfacing with existing data models and when designing new data models. You should know the basics of how relational databases work, how to design relational database schemas, as well as how to write basic SQL queries against a database.

- **Distributed Computing**

Our systems at Amazon.com usually have to work under very strict tolerances at high load. While we have some internal tools that help us with scaling it's important to have an understanding of a few basic distributed computing concepts. Having an understanding of topics such as map-reduce, service oriented architectures, distributed caching, load balancing, etc. will help you in formulating answers to some of the more complicated distributed architecture questions you might encounter.

- **Internet Topics**

This is Amazon.com, we're an online company and we expect our engineers to be familiar with, at least, the basics of how the internet works. **You might want to brush up on how internet browsers do what they do, DNS lookups, what TCP/IP and HTTP are, sockets, etc. We're not looking for network engineering types of qualifications, but a solid understanding of the fundamentals of how the web works is a requirement.**

- **Operating Systems**

You won't need to know how to build your own operating system, but you should be familiar with some OS topics that can affect code performance, such as memory management, processes, threads, synchronization, paging, multithreading, deadlocks (causes, detection, avoidance).

- **Amazon Leadership Principles**

Please make sure you understand our Leadership Principles as you will be asked to provide examples of your experiences by referencing these principles as it was applied