Technical Interview Preparation Sheet (T.I.P.S)

Technical interviews at Amazon usually comprise of questions that would be presented as a problem statement for which you would need to come up with a working solution.

The overall objective is to:

- Evaluate your **Problem Solving skills.**
    1. Think through the question and ask relevant questions so as to understand the given problem.
    2. Try and break the problem into solvable components/ Think about the programming logic that you would want to apply.
    3. Discuss the overall approach that you would like to take so as to arrive upon the solution.
    4. Work on the algorithm and relevant data structures that you would want to use so as to reach the solution.
    5. Translate your overall solution into code.

- Evaluate your expertise on **Data structures and Algorithms**.
    1. Have a good understanding of common algorithms.E.g: traversals, divide and conquer or any of the common algorithms that would you feel would help you solve coding challenges.
    2. Ability to choose relevant algorithms and being able to explain the advantages of following that particular approach.
    3. Knowing the runtimes, theoretical limitations, and basic implementation strategies of different classes of algorithms.
    4. Inner workings of common data structures and ability to compare their usage in various applications.
    5. Knowledge of run times of common application and memory usage.
    6. Check for edge cases, null cases, corner cases and validation that no bad input slips through and then present your code to the panel.

- Test your **Coding Skills :**

    1. Write syntactically correct code—no pseudo code.
    2. Ability to write scalable, maintainable, robust, and well-tested code –
    3. Production Quality Code
    4. Checking for edge cases, null cases, corner cases and validation that no bad input slips through.

Coding -Points to remember:

- Fail cases
- Time Complexity
- Dry Run before you submit the code.
- Bug Fixing
- Optimization (Ability to optimize the code)

- **Sample Coding questions :**
  - Given an integer binary tree, find the path the yields maximum sum. The path may start and end at any node in the tree. The output can just be limited to the sum (no need of printing the path). (PS and Coding)
  - Given a sorted array which has been rotated figure out how many times the array has been rotated. For ex:- 5,4,3,1,2 should give 3, The array has been sorted in ascending order.
  - Given a sorted linked list, design a solution to create a balanced binary search tree from it. (DS)
  - You are given a stream of input numbers. Print the 20 maximum elements at any point of time in the input stream.
  - Write a function to return nth fibonacci number
  - Find common element in 2 sorted arrays
  - Give a shortest path to reach from source word to destination word from given list of words.
  - LRU Cache
  - Given a stream of numbers, find the earliest non-repeating number?
  - A binary tree with Int value stored in every node is given. Find the subtree with max sum.

- Evaluate your **Design Skills – HLD & LLD**
  1. **High Level Design** skills evaluation focusses on :
     - Presenting all of the design aspects and define them in detail.
     - Describing the user interface being implemented.
     - Describing the hardware and software interfaces.
     - Including design features and the architecture of the project.
     - listing and describe the non-functional attributes like:
       1. Security
       2. Reliability
       3. Maintainability
       4. Portability
       5. Reusability
       6. Application compatibility

7. Resource utilization
8. Serviceability

2. **Low Level Design** skills evaluation focusses on :
    - Class diagrams with all the methods and relations between classes.
    - Program specifications are covered under LLD.
    - LLD describes each and every module in an elaborate manner so that the programmer can directly code the program based on it.
    - LLD will contain :
        1. Detailed functional logic of the module in pseudo code
        2. Database tables with all elements including their type and size
        3. All interface details with complete API references (both requests and responses)
        4. All dependency issues
        5. Error message listings
        6. Complete inputs and outputs for a module.

Points to remember in design Interview.

- Scalability
- Fault Tolerance
- Architecture
- Technology
- Component Wise Design
- Dependence of components
- Should know the effects if developers makes any change on one component.

Sample Design Questions:

- Design a chess game. Give me APIs and class design for the server and client module?
- Design an electronic voting machine (EVM)
- Design BookMyShow
- Post Tweet & Fetching list of tweets of all people whom X is following
- Design a Parking Lot
- Design snakes ladder problem.
- Design for Google News like eCom aggregator
- Traffic Management Signal
- Makemytrip.com
- Design a newspaper subscription service?

- Design a monitoring system
- Design a Food ordering system (swiggy,food panda etc)
- design for a system that controls lights at a traffic intersection
- How will you design a rate throttle for IRCTC?
- Explain the architecture of Drover.
- Design a reminder system. Basically design a scalable system where a user can create a reminder and specify the different medium of communications where he/she could get notified

General Preparation Tips :

- Practice coding with pen and paper (without IDEs).
- Try practicing coding questions with HackerRank, Hackerearth, geeksforgeeks, careercup.
- Prepare well on distributed systems and engineering challenges pertaining to scalability.
- Additional online resources :
  a) http://www.slideshare.net/esumit/amazon-interview-questions-28308492?qid=0995a877-3805-425f-83ce-e87d75d5e6c3&v=qf1&b=&from_search=1
  b) http://www.geeksforgeeks.org/microsoft-interview-experience-set-56-for-sde-2/
  c) http://www.geeksforgeeks.org/amazon-interview-experience-193-for-sde-1/
  d) http://www.geeksforgeeks.org/amazon-interview-experience-set-169-sde-2/

General Sample Questions:

•What do you look for when you are doing a code review?
•Why are you looking for a change?
•Initiatives Taken for things which can be improved
•Worked against tight deadlines in ambiguous situation
•Talk about a situation when you had to deliver a project/feature with a tight deadline. How did you handle and how did it impact the deliverable.
•Give an instance where you convinced your boss / senior peers about your proposal?
•Instance where you went out of your way to deliver something?
•Do you believe leadership can be exhibited in a non-managerial role?
•Talk to me about an instance where an error on your part caused major impact to your team or organization.
•Which has been the most technically challenging project so far. Let's take a deep dive
•What has been learning over the years regarding organization of code?

Note:

Please note that this is only a reference document to supplement your preparation and is by no means an exhaustive source on the interview process. Kindly use this as a framework to channel your overall preparation.