

## LEADERSHIP PRINCIPLE SESSION SUMMARY

### 1) Own that code : Culture fitness is very very important

- i) Curiosity to take interest in fixing the issues
- ii) Many Smart(really smart) peoples getting rejected after bar raiser rounds
- iii) Interview is all about cultural fitness
- iv) Communicate your approach
- v) Listen to your interviewer
- vi) Interviewer observes every behavioral skills while tech interviews
- vii) Don't ask for help directly ,you should try to google it before asking for help
- viii) Before helping anyone ensure that you're not spoon feeding ,ensure the person who you are helping is growing
- ix) You have to know which help to offer and which not to offer
- x) Properly format the questions,do HW before asking for help
- xi) Be Polite, Ready to hear

### 2) Customer Obsession

- a) Who'd be benefited from this project
- b) Talk about your customer(build for which kind of people)
- c) You should have clarity on project
- d) Your customer found a problem (like indentation in online ide) ->you took that challenge and you added a feature of indentation.

### 3) Ownership

- a) Always focus on long term goal,like (Maintainable code,scalable,flexible (versions can be changed easily,adding features)
- b) It clearly reflect in coding interviews,ask counter questions like(leading zeros,max length,valid test cases)
- c) Clarify question by asking legitimate questions,take ownership of your code

### 4) Bias for action

- a) First approach it should function anyhow(business should be working)
- b) You have to complete tasks in a timely manner,you can use some hacky ways if there are urgent needs.
- c) Act according to the situation,action depends on situation(google down example)

### 5) Deep Dive

- a) Do not just care about shallow aspects of things
- b) It also depends on time and situation

### 6) Invent and Simplify(does not apply to new grads)

- a) You don't need to complicate things
- b) Questions are framed according to level of understanding

### **7) Learn and Be Curious (Imp for new grads)**

- a) Show that you are curious to learn,tell the situations where you have learned something in very less time
- b) Requirement Gathering (test cases etc)
- c) You made effort to learn it,you have to learn lot of tech specific things (Company want you to be adoptable)
- d) Curious people listen a lot(listen with their heart),make you listening skills good
- e) Be in Present moment(don't think about the results ,otherwise you may mess up)

### **8) Hire and Develop the Best (For Senior peoples)**

### **9) Insist on the highest Standards : 2 ways of finishing the task**

- a) Mediocre way
  - i) Don't just go ahead and finish it
- b) Amazing way
  - i) Modular code,reusable components
  - ii) Useful comments (if needed)
  - iii) Good variables names
  - iv) Took care of Edge cases

### **10) Frugality**

- a) How can you achieve max using min resources(like how many machines do you need in peek days)
- b) Like reducing budget but not quality
- c) Playing with what you have,economical
- d) Optimizing your resources(don't be uncle scrooge who don't want to spend money)
- e) Like while write code don't use variables unnecessary

### **11) Earn Trust**

- a) You should not contradict yourself
- b) Think before you speak,be thoughtful,state the fact,prepare properly
- c) If you are not sure just say you don't know
- d) Fictions usually get caught
- e) **Vocally Self-critical** - Useful in life as well
  - i) Leaders are vocally critical
  - ii) Accept your mistakes
- f) Accepting mistake does not make you small
- g) Never hide your weakness (if it is not hideable)

### **12) Have Backbone: Disagree and Commit(tested lot in freshers)**

- a) Like your answer should be data driven
- b) Self confident,Proof of correctness is very imp
- c) Build intuition around your approach

- d) All this traits has to be shown with humility
- e) Humble throughout, and you should have thought of proof (take sample inputs)
- f) If you said you are 100% sure, then you should be able to implement it 100%, no u  
turn

### **13) Deliver Result**

- a) Eventual thing of project
- b) No matter how knowledgeable you are no matter how critical think you are if you  
are not able to deliver result then it's not a good thing
- c) Either it is done or it is not done, there is nothing like 90% correct
- d) Worlds perception is Boolean (You are successful or unsuccessful)
- e) You should have good delivery rate