Interviewing

"My short-term goal is to bluff my way through this job interview. My long-term goal is to invent a time machine so I can come back and change everything I've said so far."

The Basics

- Show up on time.
- Look professional.
  - Better than the average employee.
- Be awake.
- Plan ahead to make sure you know where you are going and who to ask for.

Before the Interview

- Research the position.
  - BAD: a ‘dev’ writes code, a ‘tester’ tests it, and a ‘manager’ keeps them on track.
  - Research exactly what that position means at that company.
- Study Computer Science basics
- Plan answers to ‘common questions’
- Practice!
  - It’s a lot more difficult than you might think.
  - Practice on paper and/or a white board. Don’t use an IDE.
- Do general practice problems.
  - Use online suggestion sites (glassdoor.com) and practice problems.

Computer Science ‘basics’

- Know the details about the language you will be using.
  - For Java: static, implements, extends, garbage collector, classes, methods, objects, etc.
- Know algorithms and data structures (you will be asked about these):
  - Quick sort, Merge sort, Tree traversals (BST especially for sorting)
  - Depth/breadth first search.
  - Arrays/ArrayLists/Linked Lists/Etc.
    - Insert/delete/search → efficiency of each?
  - HashMaps (or other Hash object)
    - Why it’s useful
    - Insert/delete/search → efficiency of each?
  - Properties of different variable types
    - Read the javadoc on Strings.

Technical problems to expect

- They're meant to make you think (and talk).
- Rarely will the problem be obvious/blatant:
  - "Write me a program that returns the 5th character in a string."
- The problems are designed to have more than one correct answer.
  - "If you have an array of integers, give me back a list of all integers which appear more than once."

Answering a technical problem

- Do not obsess about finding the best answer right away. (they don't care)
  - Start with an obvious solution, write it out, and then improve on it.
- TALK!!!
  - You're code is *not* important.
- Plan it out first, don't just start coding.
- Ask tons of questions, even if they may be obvious.
  - Clarify the prompt.

Non-technical questions

- They will ask about you:
  - Experiences you've had.
  - Lessons you've learned.
  - Challenges you've overcome.
  - Etc.
- Assume they have read your resume, transcript, and application.
  - Reference to it if that will help your answer.
- Come prepared with a few 'stories' about your Computer Science history.

Now it’s your turn

- Come with questions.
- Ask for advice.
- Use this time to tell them anything extra about you that hasn’t come up.
  - If you have a great story, find a way to segue to it.
- This is their time to sell their company to you.

Final Notes

- Try to stay relaxed.
  - Be a nice and interesting person, beyond your code.
- One bad interview won't end your chances.
- They want to help you, let them do so!
  - Ask tons of questions.
  - Ask for advice.
- If you've gotten to the interview, you're already doing well!
- Look back at these slides, and find what you're missing.

After the Interview

- If you don’t hear back within a week or two, send them an email to remind them of you.
- Don’t take it personally if you don’t get an offer.
- You can ask for more time to make a decision (within reason)
  - You can also ask them to hurry up if you have another time constraint.
- If you don’t like the team they place you on, you can ask for another position.
  - BE POLITE!