CSC290 Communication Skills for Computer Scientists

Lisa Zhang

Lecture 12; April 1, 2019

Agenda

- Preparing for the Future
- Course Evaluation
- Technical Interviews

Preparing for the future

How do you prepare for

- Academia (Graduate School)
- Industry (Job)
- Startup

Graduate School

There are good and bad reasons to go to graduate school.

Good reasons:

- To learn how to be a researcher.
- To see if research is right for you.
- To gain deep understanding of a subject.
- To get a job that requires an advanced degree.

Bad reasons:

- Because you don't know what you want to do with your life.

Professional vs Research Masters

Professional Masters:

- Take more courses
- Possibly an internship or research project

Research Masters:

- Take some courses
- Mostly conduct your own (publishable) research
- Write a thesis

Goals behind Professional vs Research Masters

Professional Masters:

- To get a job that requires an advanced degree.

Research Masters:

- To see whether you want to continue to a PhD.
- To increase the total sum of human knowledge.

Graduate School

To successfully apply to graduate school, you will need:

- Good grades
- Letters of reference from three referees

Reference Letters for Graduate School

If you might want to do research, think about your letters early.

In order for a prof to be able to write a good letter for you, they should know who you are!

A letter that simply says that you did well in their course is not helpful for the admissions committee.

What can you do?

- Get to know the profs whose courses are aligned with your interests.
- Have conversations about the research area in office hours.
- Participate in course message boards.
- Participate in extra-curricular activities.

Your profs want to write a good letter for you, so get to know them early.

Asking for a reference letter

- All your profs had to ask for **many** letters to get to where they are!
- In your email, ask whether they would be able to write a **strong** letter of recommendation
- Remind them of what you did and what you want to be included in your letter:
    - e.g. that you participated in certain activities, grade you got in their course

Summary: Grad School

If you might want to do research:

- Think about what areas you find interesting.
- Approach professors teaching courses in possible areas.
- Participate in those courses by attending office hours, course message boards, so that those profs know who you are.
- Show interest, initiative, and good communication skills.

Your request for a letter should not be the very first time you speak to a prof.

Industry

Yes, the old adage “It’s not what you know, it’s who you know” is true:

- Keep in touch with your classmates.
- Get to know students entering the work force a couple years before you.

People want to know interesting people

The flip side is that people are interested in knowing interesting people.

- People who do interesting things.
- People who write about the interesting things they do.

This is why blogs can be powerful tools!

- Take time to do interesting (CS) things, and write about them!
- Share your work on HackerNews, Reddit, etc.

Getting Hired

- Reaching out to people (e.g. with a cold email) is the modern day equivalent of “knock on doors”.
- LinkedIn profile can be helpful.
- Don’t burn bridges, not even with classmates!

Try to understand the hiring process at companies you are applying to.

If you don't get into POST

- Your degree (or lack thereof) does *not* define you.
- My own degree was in Pure Math.
- I worked with team leads who did *not* go to university at all!

Summary: Industry

- Do interesting things, communicate them, and get to know people.
- Let your intentions be known in your community.
- In CS, your skills and abilities matters more than a degree.

Startup

Startups are companies designed to *scale quickly*.

- Not all interesting ideas are meant to be startups.
- But there is so much resource around startups, that things that shouldn't be called "startups" ends up being called that.

Are you interested in a *product idea*, or the idea of growing a company quickly? (You need both.)

Think long term

- Startups can be a 5-10+ year commitment.
- It will be an emotional roller coaster.
- You'll need to wear many hats, even the ones you don't like.
- Survivorship bias:

Think long term

- Startups can be a 5-10+ year commitment.
- It will be an emotional roller coaster.
- You'll need to wear many hats, even the ones you don't like.
- Survivorship bias: we only hear about the successful startups, not the failures.

Skills

- Learn to communicate.
- Learn to identify and talk to potential.
- Learn to be okay with rejections.
- Go beyond coding: coding is only a small percent of what makes tech startups successful.

Resources

- Creative Destruction Lab
- NextAI
- Hatchery
- More at http://entrepreneurs.utoronto.ca/accelerators/

Summary: Startup

- It is a difficult road.
- Build soft skills and other non-technical skills.
- Use the resources available.

What if you don't know what you want to do?

That's okay! You don't have a lot of information yet. Your main goal should be **getting that information to help you decide**:

- Explore! Explore different options.
- Be okay with spending time and effort on things that are only "interesting".
- Be passionate about what you are working on, even if you're not 100% sure.

I believe that passion comes before success, *not* after, and that discipline is more important than motivation.

What I don't recommend

Stay in school an extra year to "find oneself".

- If four years didn't give you the information you need, why would another year be different?

(Note: there are other valid reasons to stay an extra year)

Finding oneself

- Putting yourself in different situations to see how you would respond.
- Challenging yourself.

In all cases...

Soft skills are extremely important, and you *will* write a lot:

- Academia: write research papers, present papers, communicate with collaborators, ...
- Industry: write project proposals, commit messages, bug reports, documentation, feedback, ...
- Startup: write grant proposals, emails, contracts, user guides, documentation, ...

So in either case:

- Practice writing: blog?
- Network with people: get to know your classmates, profs, members of the community
- Make cool things – that's probably why you interested in Computer Science.

Course Evaluations

I wish you success, whatever you choose to do.

Interviews

Types of Interviews for Software Developers

- **Behavioural**: Assess soft skills
  - e.g. explain how to handle a situation
  - e.g. share example of situation where you . . .
- **Design**: Assess high level design skills
  - e.g. how to design a parking lot?
- **Technical**: Assess low-level programming skills
  - e.g. how to reverse a linked-list?

Behavioural Interview Example

- What would you do if you and your coworker have conflicting opinions about something?
- What would you do if there is a deadline, but your coworker and boss have conflicting opinions about how to proceed?
- Can you tell me about a situation where you had to work under pressure?

Behavioural Interview DOs and DON'Ts

DO:

- Make sure you understand the question
    - repeating the question gives you time to think
- Think before giving an answer
- Share the credit with others when they are due

DON'T:

- Appear sarcastic or bitter
- Speak poorly of your former coworkers
- Talk for too long without giving your interviewer a chance to interrupt

First Impressions

- Appear clean and professional
  - Not necessarily a suit, depending on the company
- Have a firm handshake
- Be nice to everyone, including the receptionist and non-technical folks
- Arrive a little early (5-10 min)

Technical Interview

Most of your interviews will fall under this category

The goal of the interviewer is to check...

- if you understand the common algorithm techniques
- how you think
- whether you can effectively communicate your thoughts and ideas

You will be expected to write code, either on a computer or on a whiteboard

Typical Technical Interview Format:

- First 5-10 minutes:
  - introduction, a little about the company/team/role
- Middle 20-40 minutes:
  - one or more technical questions
  - usually (but not always) the interviewer prepares more questions than there is time for
- Last 5-10 minutes:
  - reserved time for candidate questions

Types of Questions

- Usually small algorithms questions
- Learn CSC263 materials well

Example:

- Given a string consisting exclusively of opening and closing brackets, check if the brackets are balanced.
  - For example “()(()(()()))” is balanced, but “((”, “())” and “()())(” are not.
- In many languages, the integer type can only store integers up to a certain size. How can we create a “big integer” representation in a language like that? How do we add two integers represented in the method you chose?

Solving Interview Questions

- Start by **making sure that you understand the question**
- Ask if there are certain assumptions that you can make
  - e.g. "Can we assume that there is only one type of bracket?"
- Make sure that you are solving the right problem, and give yourself time to think

It is important to **think out loud** so that your interviewer can see how you're thinking

Your First Idea

- Don't try to solve the problem in one shot, unless you see the solution right away
- It is okay to start with a slow solution, or to draft out several ideas before committing to one
- Sometimes, the slow or naive solution is the one that the interviewer is looking for

If you can't come up with an idea. . .

- Come up with a **simpler problem** to solve first
    - Sometimes, thinking about a simpler problem will help you think of the actual solution
- Talk about ideas that **won't** work, and where the idea fails
- Talk about a **related problem** that you know how to solve, and why this problem is different
- Give the interviewer opportunities to help you

Your Interviewer...

Your interviewer wants you to succeed, and might help lead you to the right idea.

The way you listen to the interviewer's suggestions also impacts their assessment.

Let's conduct some interviews!