

# HOW TO GET A JOB IN CS

## Putting your CS-214 skills to work

*“This lecture is going to be HIRE 🔥” — Anonymous CS214 student*

Modularity ✖

Correctness ✓

Real-world engineering 🔧

CS-214 W13 SE – 2025-12-08

Clément Pit-Claudé

# Quick announcements

## Last week of webapps!

Time for polishing.

Check your slot!

Prepare for the checkoff.

## Exam format

Small, medium, large exercises

Somewhat too long, on purpose

## Final exam dry-run

Wed 17th

## Last chance to test setup

Full, 3h exam in real conditions

# Today

## Getting a job in CS

- Identify the key parts of the job-search process.
- Build a compelling CV and portfolio.
- Structure answers to skill- and situation-based interview questions.
- Avoid common job-search pitfalls.

- Motivation: Why now?
- Building a profile
  - Networking
  - Portfolio
- Making opportunities
  - Activating your network
  - Reading job ads
- Applying
  - CV & top CV sins
  - Cover letters
  - Recommendations
- Passing interviews

***“Why do I have to worry  
about this now?”***

*— CS214 student who would prefer to talk about futures*

# REASONS TO PLAN AHEAD

- Looking for internships
- Applying for SA/AE positions
- Applying for semester projects
- Applying for the MS fellowship
- Studying abroad
- Founding a startup

Plus, things take time:

- Training for interviews
- Building a profile
- Building a network

# THE JOB SEARCH TIMELINE

## 0. Build skills, a profile, and a network

From now to the end of your career

## 1. Search for opportunities

A few months

## 2. Apply and get invited to interview

A few weeks; aim for 10-20% hit rate

## 3. Interview and get an offer

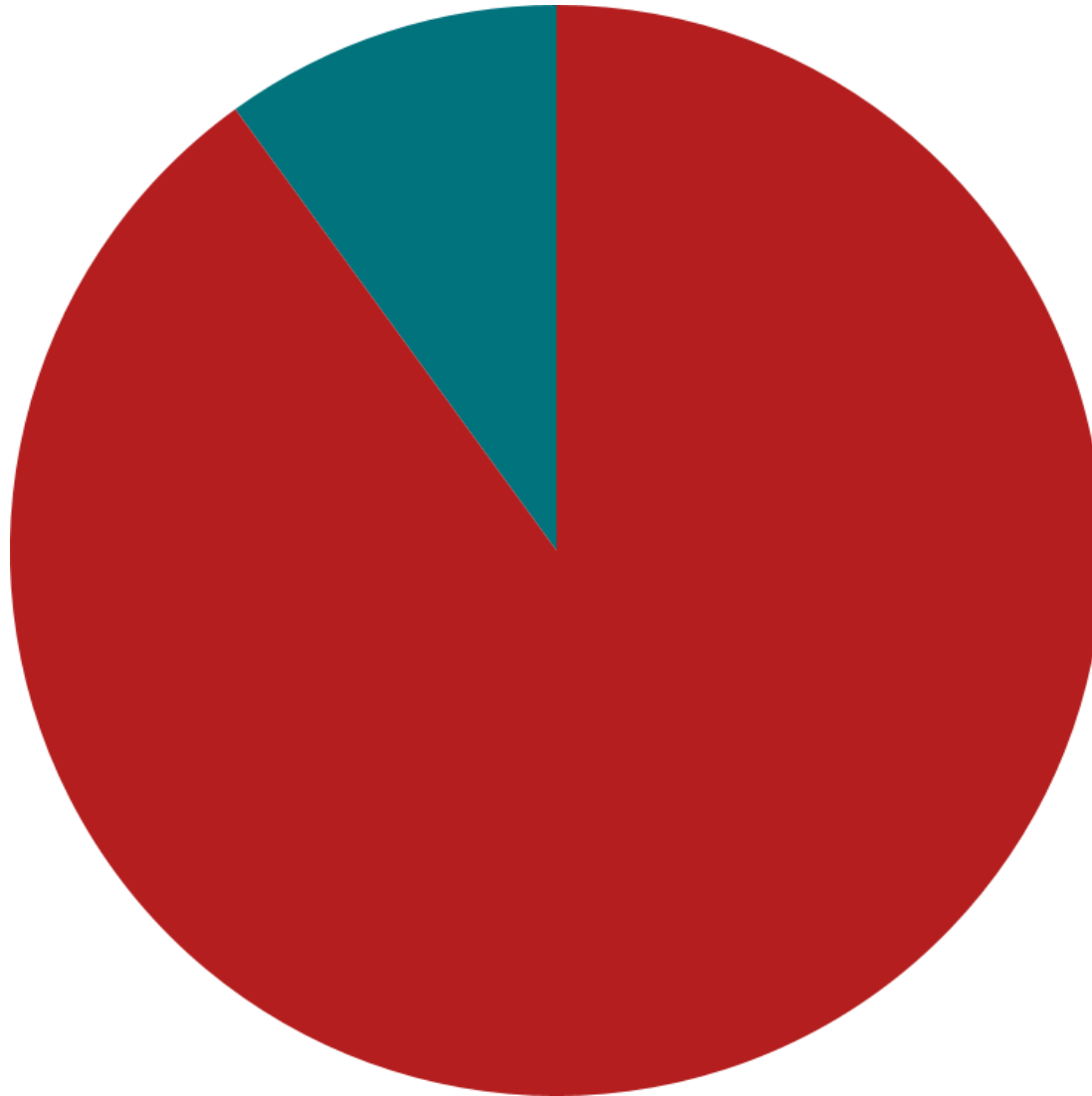
A few days to a week of interviews

1-2 weeks to get an offer; may ghost if negative

PART I

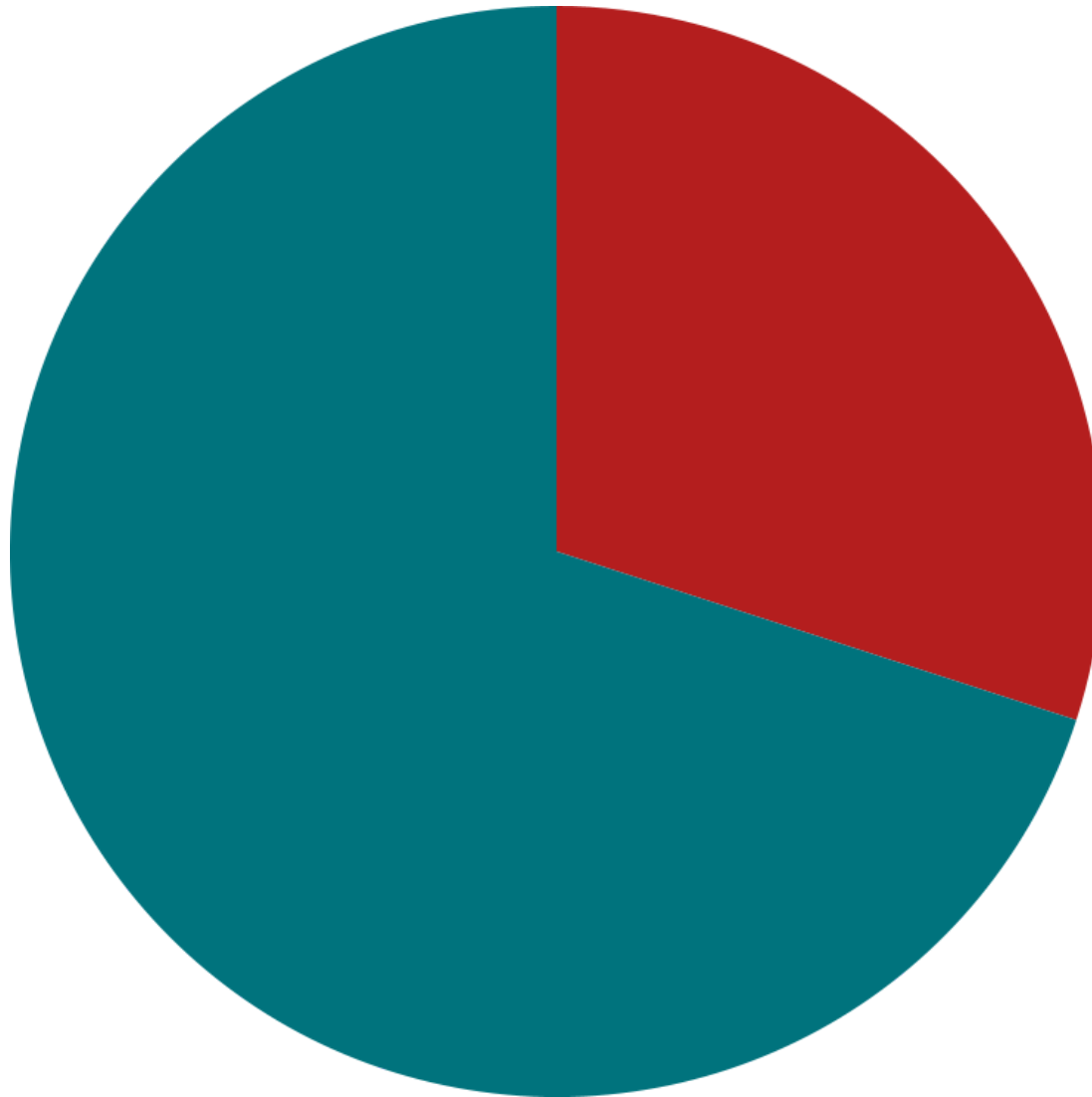
# **FINDING AND MAKING OPPORTUNITIES**

# THE TYPICAL VISION OF THE JOB MARKET



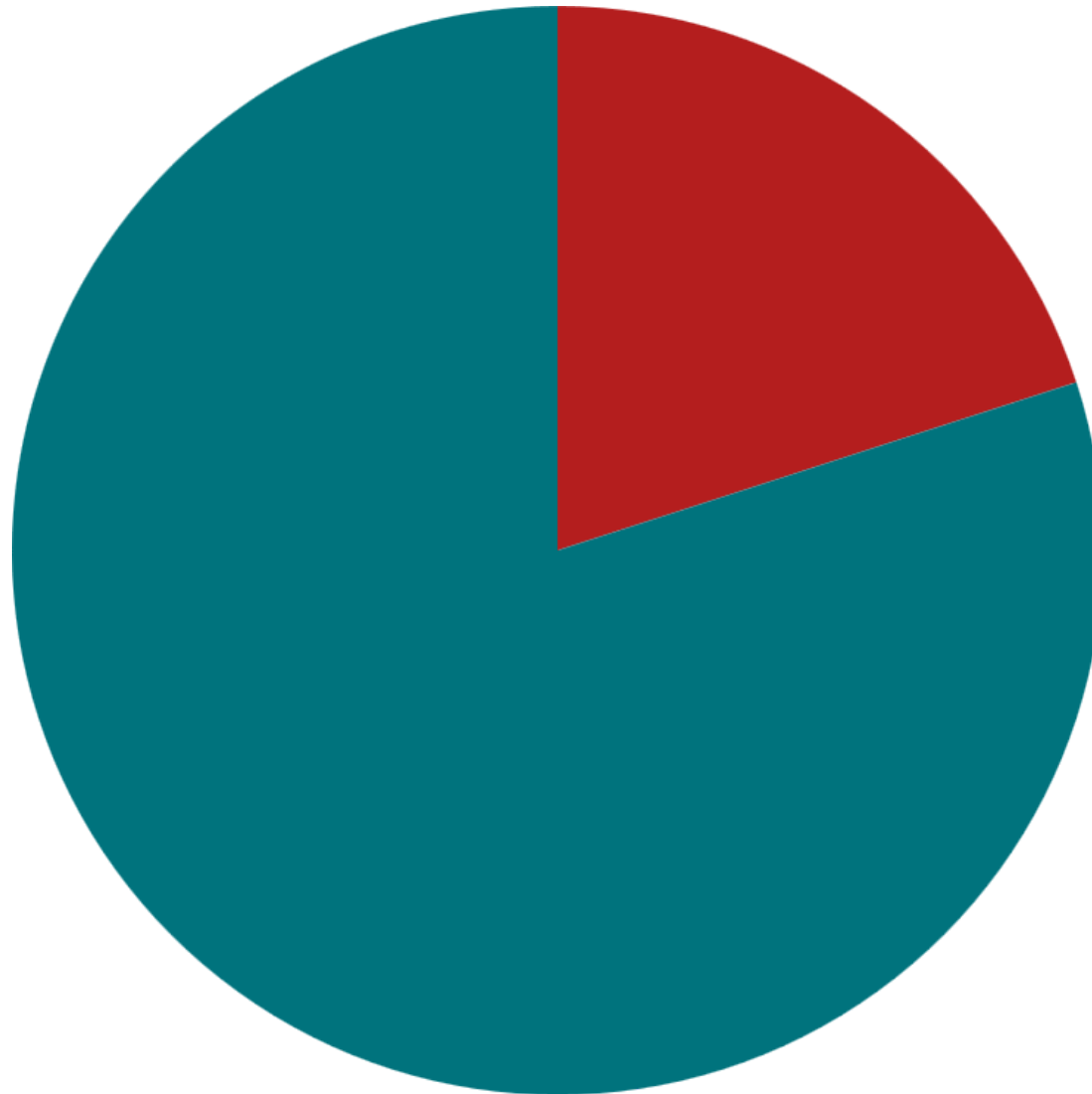
- Advertised online
- Other

# THE REALITY OF THE JOB MARKET



- Advertised online
- Other

# HOW POSITIONS ARE ACTUALLY FILLED



- No referral
- Personal or professional connection

# DO REFERRALS MATTER? WHY?

## Referrals matter a lot

- Companies are overwhelmed with applications
- Interviews are very costly
- CVs can be fabricated, candidates lie
- The market is bad

## You need a **meaningful** referral

- Putting your name into a random system: **pointless**
- Connecting you to a team looking for your profile: **priceless**

# MAKING OPPORTUNITIES

- **Be visible (see next section)**

Keep a public and active profile

- **Look for open positions, even only vaguely related.**

Job posting websites, EPFL internship site

- **Look for relevant companies (skills-needs alignment)**

Don't waste time applying to irrelevant jobs!

- **Put the word out! Talk to friends, colleagues, relations**

Ask for tips, recommendations, experiences

- **Reach out to contacts — leverage your network!**

Caveat: Power of network  $\propto 1 / \text{rarity}$

# READING JOB ADS

- Check overall relevance / skills
- Don't obsess over specific details



**Sebastián Ramírez**  
@tiangolo

Follow



I saw a job post the other day. 📄

It required 4+ years of experience in FastAPI. 🤔

I couldn't apply as I only have 1.5+ years of experience since I created that thing. 😂



SOFTWARE

192 🗨️

**IBM job ad calls for 12 years' experience with Kubernetes – which is six years old**



# **DO NOT SPAM**



- **Do not send mass emails**
- **Do not resend an email 4 times**
- **Do not ask 5 labs for a project at the same time**
- **Do not sound desperate**
- **Respond quickly**



# **APPLY SMART**



- **Do not apply to 50 job postings**

EXERCISE

# MAP YOUR NETWORK

PART 2

# APPLYING

# WRITING A CV

## A CHECKLIST

- One page
- Clearly structured
- Clear profile and strengths
- One-sentence objective / intro
- Projects, portfolio with links
- Important info stands out
- You* stand out

Good form matters. Content matters much more.

**Tip:** Include a TL;DR.

# WRITING A CV

## AN EXAMPLE

### Good? Bad?

#### Devops

Implemented streamlined pipeline based on state-of-the-art open source libraries to produce a clean, robust, scalable system with fast response times. Deployed system and ensured reliability and user satisfaction.

### It's bad. The following is much better:

#### Junior DevOps engineer, Foo Corp

Replaced a bespoke sales-monitoring system with a grafana-based dashboard, reducing maintenance costs by 0.3 FTEs and shrinking report-generation time from 4 hours to 2 minutes.

# WRITING A CV

## COMMON SINS

- ✗ Technobabble
- ✗ Vague terms, fluff
- ✗ Unclear responsibilities
- ✗ Unclear contributions
- ✗ Missing names / institutions
- ✗ Missing keywords
- ✗ Too short
- ✗ Too long, pointless padding
- ✗ Methods over results
- ✗ Not following country norms
- ✗ Bad fonts, layout, grammar
- ✗ Lies and exaggerations
- ✓ Be understandable
- ✓ Be specific, have numbers
- ✓ Say what you role you had
  - ✓ Say what you did
- ✓ Highlight relevant places
  - ✓ Don't get filtered out
- ✓ Highlight relevant experience
  - ✓ Stop boasting, stay relevant
    - ✓ Say what, not just how
- ✓ No photo in most countries!
  - ✓ Look good
- ✓ You'll get caught. It'll suck.

## EXERCISE

# WRITE A CV BULLET POINT

### Infrastructure engineer, *Software Construction*, EPFL (10h/w)

Developed and maintained course infrastructure for ~400 students.

- Ensured compliance with data-retention policies by developing backup scripts for student home directories (`rc1` one to retrieve data from NFS, `Btrfs` to create low-overhead snapshots, and `systemd` timers for periodic execution).
- Reduced risk of dependency issues during exams by creating an SBT plugin to check for configuration inconsistencies in CI.

## EXERCISE

# WRITE A TL;DR

### Objectives

- Start a project from scratch
- Use advanced math
- Develop new algorithms

### Objectives

- Join an existing project
- Work with a large codebase
- Try low-level programming

### Top courses

- Algorithms: 5.75
- Advanced compilers: 5.5
- Software Construction: 5.25

### Top courses

- Distributed systems: 4.5
- Software Construction: 4.25
- Computer Systems: 5

### Top skills

- OCaml
- Linear algebra
- Compression algorithms

### Top skills

- C, Rust
- Git
- Debugging

# COVER LETTERS

- Half a page
- Have a story about yourself
  - How you got where you are
  - Why you want to be there next

← **Customize this part**

## Sins

- ✗ Pointless / generic blather (Modern version: ChatGPT)
- ✗ Vacuous / unsupported statements
  - *“I am highly motivated”*
  - *“I am convinced your company is a great match”*

# COVER LETTERS

## Don't bullshit: 2025 version



You

I'm applying for a paid position as an undergraduate teaching assistant in a Software Construction course, please help me write a cover letter email to the teacher to get the job.



ChatGPT

Subject: Application for Undergraduate Teaching Assistant Position in Software Construction

Dear Professor [Last Name],

**I hope this email finds you well.** My name is [Your Name], and I am writing to **express my enthusiastic interest** in the Undergraduate Teaching Assistant position for the Software Construction course. I am currently [mention your academic year] **majoring in [Your Major]** at [Your University], and **I am confident that my academic background, passion for software development, and strong interpersonal skills make me an excellent candidate for this role.**

**Throughout my coursework, I have gained a solid foundation in software development principles,** including [mention specific skills or programming languages relevant to the course]. I am particularly drawn to the Software Construction course **because of its emphasis on practical application and real-world problem-solving,** which aligns perfectly with my **academic and professional aspirations.** **I am eager to contribute to the learning experience of my peers by sharing my knowledge and assisting them in mastering the key concepts covered in the course.**

My previous experiences, including [mention any relevant experiences such as internships, projects, or coursework], **have equipped me with effective communication and collaboration skills.** These skills, combined with my **ability to break down complex concepts into understandable components,** will enable me to support students in **comprehending challenging topics** and completing their assignments successfully.

**As an aspiring educator, I am excited about the opportunity to work closely with you** and the students to create a positive and engaging learning environment. **I am confident** that my dedication, technical expertise, and commitment to fostering a collaborative atmosphere will contribute to the success of the Software Construction course.

Thank you for considering my application. **I would be thrilled to further discuss how my skills and experiences align with the goals of the course during an interview.** I am available at your convenience and can be reached via email at [Your Email Address] or **by phone at [Your Phone Number].**

**I appreciate your time and consideration and look forward to the possibility of contributing to the success of the Software Construction course.**

Sincerely,

[Your Full Name], [Your University ID/Class]

**[Your Contact Information]**

# RECOMMENDATION LETTERS

- Chose relevant people
- Make sure they know your name
- Give them something to write about
- Ask whether they can provide a *great* recommendation letter

PART 0

# **BUILDING YOUR PROFILE AND YOUR NETWORK**

# YOUR PROFILE

## A ROADMAP FOR THE NEXT 3.5 YEARS

Show independence and initiative, stand out, have fun. Some ideas:

- **Free software contributions**

Contributing is harder than building from scratch. Many projects have onboarding guides.

- **Personal projects, hackathons, junior enterprise**

Most students stick with just coursework — Webapp fits!

- **Semester projects in labs**

A must if you want to do R&D or a startup

- **Paid student assistantships**

Great for your network

- **S4S, Fréquence banane, IT for a club**

Springboard to get an assistantship

- **Volunteer / nonprofit work**

Make it relevant

- **Language training**

English, then French, then German

- **Course projects and assignments**

Only if they are good

# YOUR NETWORK

## A ROADMAP FOR THE NEXT 3.5 YEARS

### At school

- **Attend events**  
Talks, career fairs, hackathons, conferences
- **Find group hobbies**  
Hiking, gaming, music, anything
- **Have a study group**  
Bonus: it helps with grades
- **Get involved in mentoring**  
Seek it and give it: you may eventually depend on your mentees
- **Get involved in committees**  
Serving your community always pays off
- **Move around!**  
Study abroad, do internships

 **Networking is mostly one-on-one.**

**Introverts can be great at networking.**

# YOUR NETWORK

## A ROADMAP FOR THE NEXT 3.5 YEARS

### Online

- **Have a LinkedIn**

That's where recruiters are

- **Keep in touch with friends**

From high school, clubs, summer camps, military service, ...

- **Join mailing lists, discord, IRC**

But be careful with contributions under your name. Stay professional.

- **Publish tools and code**

Not just for your portfolio: also helps meet people

- **Blog, follow social media**

Get a literal pressbook

## EXERCISE

# MEET NEW PEOPLE IN THIS ROOM

Introduce yourself to three people you recognize but haven't really spoken to.

- Name, section
- Your strengths (courses, techniques)
- One uncommon skill or interest (work-relevant!)