



Technology Recruitment

Many thanks to the Swarthmore alumni working in various tech positions for sharing their insight below.

Roles

Software Engineer - develop/design software, write code, fix bugs, write tests (software that catches bugs), give tech talks, review other people's code, project planning

Quality Engineer - testing, try to break code

Performance Engineer - optimizing software, performance tuning and analysis

Release Engineer - ship large amounts of software

Systems Engineer - interdisciplinary, includes management, design and manage complex systems over their life cycles

Security Engineer - network/systems security

Data Scientist - optimization of large data sets

Data Analyst - analyzing data

UX Designer - user experience, visual, build great experiences for humans

Tech Doc Writer - user documentation

Product Manager - connect customer to engineering team, market research, embedded in engineering team

Project Manager - organize the work of engineering teams

Trends

- Coding is extremely agile; your customers and industry change constantly
- Work is iterative and extremely collaborative; you will never be able to produce by yourself what you can with a team
- Emphasis on Minimal Viable Product (MVP): build small, ship, get feedback, make iterations, continue to grow
- Coding is fast-paced: continual learning, need to adapt and stay abreast of changes in industry

Large vs. Small Companies

Large

- **Description:** mature technical processes, will learn by osmosis, interesting architectural problems and a lot of research going on, opportunity to attend many technical talks, training programs and clear career paths
- **Recruiting:** large, substantial recruiting teams; majority of university recruiting in fall (FT and internships); recruit on campus, at conferences and career fairs and on their university recruiting site; network with Swarthmore alumni (through LinkedIn and Swarthmore alumni directory)

Startups

- **Description:** funded by angel investors, VCs, Series-A (first round of funding after angel investors), IPOs. Generally smaller but potential for massive growth. More time to code and

wear multiple hats (can take initiative and gain experience in many different functions). High risk, less stable but potential for massive growth

- **Recruiting:** smaller recruiting teams (if any), timelines vary but are usually later than large companies, recruit online or through their own networks, use the Swarthmore alumni network for referrals

A note about the Gig Economy: Read about retirement saving in our *Life After Swarthmore* guide. Read industry reviews in Vault (log in through your Swarthmore Handshake account). Also consider sources for health insurance - are you getting it through your employers or on your own. If it's a short term project, talk with your supervisor about a recommendation after the project ends.

Alumni @ Successful Tech Startups and at other tech companies include (find them on LinkedIn for their most updated employment and career path):

Airbnb	Donna Boyer
Ancestry	Evan Wittenberg
Care.com	Matt Coddington, Josh Smift
Cloudera	Haley Most
Dropbox	Amandine Lee
Opentable	Miles Skorpen
Pivotal	Marie Cosgrove-Davies
Spotify	Kenny Ning
Square	Dan Rice
Stitch Fix	Cathy Polinsky
SurveyMonkey	Chuck Groom
Wayfair	Saurav Dhitel

Also, check LinkedIn's **Career Insights** feature to learn about the career paths of alums in specific locations, organizations and fields. You can search by these categories at <https://www.linkedin.com/school/swarthmore-college/alumni/>

Swarthmore's own Online Alumni Community is a great resource for this information, as well. Read Swarthmore's [Networking Guide](#) for advice and information about contacting alumni.

For example, you will find Swat alums are working in tech at these organizations:

Guidepoint	Spotcloud	Uber
Stripe	Verily	Waymo LLC
Flipboard	Starry	EPIC Systems
Reddit	RISE	

Networking

- **People:** alumni, faculty, conferences, Career Services
- **Online:** LinkedIn, Facebook, job sites, conferences, Github

Resumes

- **Experience:** project description, what you did, why you did it, what technologies you used, what was complex/novel, how you tested it, impact/results. 30-40% of resume should focus on projects. Add keywords that companies search on (see job description).
-

Workshop on Technical Resumes, conducted by Google representatives at Swarthmore

General Guidelines:

- focus on **results** and **impact**, listing **metrics** that measure impact rather than long job descriptions
- add LinkedIn and GitHub links to heading of resume; companies will review both so be sure your sites are professional
- include any open-source project links
- register the domain of any personal website with GoDaddy.com so it won't expire
- make sure any embedded links in the resume are functional
- resume length: one page per 10 years of experience
- don't be afraid to show who you are, e.g. extracurriculars that connect to the organization's values
- use bullet points and be concise
- cover letters aren't necessary for Google

Education Section:

- list key courses, including Discrete Math, Data Structures, Algorithms, Advanced Algorithms, Databases, Machine Learning, Artificial Intelligence, Distributed Systems, Linear Algebra, Vector Calculus and other math classes
- include important projects and group research you have completed in your courses; can include under Education or in a separate Projects section
- can include prizes won at hackathons in an Awards section within Education (or in a separate Awards section)

Technical Skills:

- list immediately after Education in its own section
- list object-oriented programming languages (e.g. Java, JavaScript, C++, C, C#, Python, Swift) in order of proficiency from most proficient to least proficient
- feel free to indicate level of proficiency, e.g. Proficient in Java; Familiar with Python
- be ready to conduct your interview in any of the languages you've indicated you're proficient in; you don't need to be proficient in every language on your resume, but you should know at least one well
- can include foreign languages and other technical skills in this section

Experience Section:

- showcase your academic research, open-source projects (e.g. *Contributed x line of code to y open-source project* and include a GitHub link), mobile app & web development, hackathons/coding competitions, leadership in student groups, participation in programs like Google Summer of Code
- publications, presentations, papers and patents are great to include but not needed if you don't have
- the first bullet points under an experience is typically your thesis statement, i.e. what you did
- the next bullet points describe your impact, quantified if possible; your impact indicates that without you, this wouldn't have happened, e.g. *I accomplished x by doing y, measured by z*

Interviews (see list of alumni mock interviewers at end of document)

- **Phone screens:** What's your favorite class? Why did you decide to study CS? What websites do you like? How do you learn new technology? What's the most interesting project you've worked on? Why? What technologies did you use? Did you work by yourself or with a team? How did you divide up the work? What did you personally work on? Did you have any design disagreements? Have you ever used any source control

system? How did you test your project? What was the result/impact? What would you add if you had more time? How would you scale it?

- **Technical Questions:** *Data structures* (arrays, linked lists, hashmaps, trees); *Algorithms* (know at least one sorting algorithm - bubble, merge, quick); Big O notation, best/worse average analysis; *OO design* (objects vs. attributes) *Databases* (create a schema for...write a query for...inner/outer join, tools to optimize a slow query), *Frontend* (design a webpage for...What javascript libraries have you used? Explain css box model. How do you prevent cross-site scripting-XSS)
- *Design Patterns: Elements of Reusable Object-Oriented Software* by Gamma, Helm, Johnson and Vlissides an excellent resource
- [Steve Yegge's 5 most essential phone screen questions](#): 1) Coding (write simple code with correct syntax in C, C++ or Java) 2) OO design (define basic OO concepts and model a simple problem) 3) scripting and regexes (describe how to find phone numbers in 50,000 html pages) 4) data structures (demonstrate basic knowledge of most common data structures) 5) bits and bytes (answer simple questions about bits, bytes and binary numbers)
- **Your Questions:** What made you decide to work here? Where did you intern or work before? What project are you working on? What technologies do you use? What's the release cycle like? How does your team work with other groups? Who are your top customers/users? How fast are you growing? What are your biggest challenges? Who are your biggest competitors?
- **Full Day Interview:** 4-6 45 minute interviews; each company and manager has different styles; 1-2 people will meet with you in each slot. Focus areas can be data structures, algorithms, design, databases, frontend technologies, logic puzzles and problem solving
- **Whiteboard Questions:** stay calm, listen carefully, ask questions if you don't understand, solve simplest solution first, check your code, talk about efficiency/trade-offs/improvements, iterate, think out loud/have a conversation
- **Pair Programming:** two people in front of one computer, collaboration is key, think out loud and throw out options before storming ahead, check your code, write tests, talk about efficiency/trade-offs/improvements, be the person that you would want to work with
- **Take Home Test:** read instructions carefully, ask what's most important to the tester and prioritize if timed, add your name to the code, document, add test code
- **Culture Fit:** Do you fit the values of the team? Show interest and enthusiasm; passion for technology; knowledge of company; demonstrate you're a fast learner, hard worker and have motivation/drive
- Interview is time to talk about job and opportunity, not \$

Offers

Hiring decisions are generally made by a development manager; timing depends on number of openings and candidates and generally a response can be made quickly. If you don't get an offer, it could be you're a good company fit, but not in that role; they have multiple strong candidates; they have headcount/budget issues; or the manager simply got busy and hasn't made a decision (following up is a good idea).

After you get a verbal offer, ask questions about offer, benefits, compensation and equity. It never hurts to ask for more, based on assets you'll bring to the team!

Most companies don't have a huge variation in what they offer college hires. If you were a summer intern and they're asking you to return, they'll be putting together a competitive offer. They know from your internship performance that you'll be successful so they will want you to accept. If you want to get the absolute best possible offer, your best bet is to have a competing offer from another similar or higher tier company: Amazon, Google, Facebook, Apple, Microsoft, Airbnb, LinkedIn, Salesforce, netflix, etc. This could potentially increase your offer by 5-20K -- but it will take a lot of

time to do, is not a guarantee and it might not be worth it to you if you know you want to work at the organization offering you employment and have a lot on your plate at Swarthmore.

Compensation may include base salary, bonus and equity options. Equity can mean stock options or RSUs:

- Stock options allow you to buy stock at a future date at the price it is today, but you will vest over time and can't access immediately
- Restricted Stock Units are actual stock you receive, must pay tax when sold

Multiple Offers

If you want to get an offer from another company, write to an alum at a few of the companies you're interested in, send them your resume and say that you have a pending offer from ___ but you have always admired company x and would really like a chance to interview with the company before you accept the offer. If their recruiting team is on top of things, they will fast track your application and interview process so that you can decide with both offers together.

Alumni can provide practical advice on negotiating with multiple offers.

- Be transparent
- Let recruiter know you're excited by the offer
- Ask for more time if needed
- Figure out what you want
- Give yourself a deadline

Negotiation Script

If you decide not to get a competitive offer but don't think your offer is really competitive or want to try to negotiate more, you can say something along the lines of "I really want to work at ___ and I'm very excited to receive this offer but the salary is lower than I expected. I was hoping for something in the range of \$x-yK. Is there any room for negotiation?" Then just pause and wait to see how they respond. You don't have to accept on the spot - you can be gracious but say that you need some time to think about the offer. Remember that employees earn money because they add value to an organization, so you should respectfully reiterate how you will benefit the organization.

*If you want to negotiate your offer, the big companies are more likely to budge on a signing bonus or increased equity rather than base salary.

Equity Valuation

Consider how much funding any pre-IPO company has obtained; it may be hard for employees to see much up-side to the stock. It is so hard to predict the market. Put more weight into whether this is a job where you think that you can learn and grow as a professional and make an impact and less on whether you will make millions on the stock. If you have a good boss and a good project, you'll set yourself up for a great career in the tech industry whether you decide to stay at that company for a longtime or if it's just your first company of many.

If you want to try to make a guess at the value of a company, you look at the current valuation and make an estimate on what you think its potential upside to be. For example, as of September 2017 Uber was valued at around \$68B in the last funding round. Benchmark (major Uber investor so very biased) says that Uber could comfortably be worth \$100B in the next few years. If you believe that number, that would be an increase of 47% in valuation. Assuming that most of the stock is structured equally and that there are not a significant portion of preferential stock, you could multiply your

shares times 47% to get a sense of upside. So if you are given 1000 shares at a valuation of \$10 and the value goes up to \$14.70 in a few years, you would make \$14.70 for each share netting out to \$4,700.

Most private companies are fairly tight lipped about their equity cap table but there are a few things that you should ask to be able to look at the valuation. Don't be surprised though if they can't or won't tell you the answers to these:

- What is the current RSU price/value of the options?
- How many outstanding shares are there? How much dilution do they generally have each year to fund employee grants?
- What was the internal valuation of _____ at the last option pricing?

If they won't answer these questions, I would ask them to walk you through how they would value the equity grant and what they see the potential up-side to be. Note, your shares are granted and priced at the first board meeting after you start. With a company as dynamic as many in this industry, a lot can change between now and the near future so all of your calculations may change significantly between now and then.

Attitude Counts!

- It's a small world
- Reputation is important
- Be trustworthy and professional
- Don't renege. Do not accept an offer and then rescind if you get a better offer later. It's super bad form and has the potential to impact your reputation. If you accept at a company and then later hear of another college student with a higher offer at that company, it's not unreasonable to email the recruiter and say that you are friends with someone who got a higher offer and ask if there is any room to increase the salary. Worst they can say is no.
- Inform other companies you've interviewed with and any alumni you've networked with after you've accepted an offer. Keep the door open for future opportunities.
- You're in demand...but don't be arrogant about it

Alumni Technical Mock Interviewers

The following alumni have generously offered to conduct mock technical interviews with students. Students are encouraged to reach out directly to the alumni to ask for assistance.

- Alums' job titles are listed here for students to determine which person would be the best to contact.
- Please also research these alums on LinkedIn before contacting them to help determine whose experience might be most insightful and to learn more about their work.
- Please be professional in your outreach/request and remember that the alumni are also very busy. The [Networking Guide](#) in Career Services has helpful advice for introducing yourself to alums.
- Know that there are many more alums who are willing to help students. Check the [Alumni Online Community](#) and [LinkedIn](#) with your desired search criteria.

Alex Rothenberg '92

Software Architect at Iora Health
alex@alexrothenberg.com

Micah Harkins

Amazon Web Services
Software Development Engineer I

micah825@gmail.com

Jacob Mattison, '94

Senior Software Engineer, Technolutions, Inc.
jacobmattison@yahoo.com

Matthew Katinsky '87

Senior GIS Programmer
Schlosser Geographic Systems, Inc.
mkatinsky@gmail.com

Anthony Manfredi, '07

Software Engineering Manager, Transfix.io
amanfredi@gmail.com

Demetrios Karis '74

Ph.D. in Experimental Psychology
Working as a UX (user experience) researcher
Instructor at Bentley University in Human Factors in Information Design Department
Company: Karis User Experience Evaluation
Previously at GTE Labs and Verizon
demetrios.karis@gmail.com

Alex Benn '08

Senior Software Engineer, Google
abenn1@gmail.com

Max Parke '11

Software Engineer, Google
mparke@google.com

Madison Garcia '14

Software Engineer, Google
madisong@google.com

happy to offer information and other support to tech students, but not mock interviews

Andrew Ruether '94

Head of Academic Technology Support
aruethe2@swarthmore.edu

Peng Zhao '15

Software Engineer, Google
pengzhao1993@gmail.com

Sample resume on next page

Tommy Technology

(555)-111-2345 | TommyTech@gmail.com

EDUCATION

Swarthmore College Swarthmore, PA *Bachelor of Arts in Computer Science & Economics (GPA: 3.5) May 2019*

- *Honors & Awards:* Ron Brown Scholar, Phillips Evan Scholar, and Ronald McDonald Scholar
- *Relevant Coursework:* Data Structures and Algorithms, Software Engineering, Artificial Intelligence, Computer Security, Compilers, Single-Variable Calculus, Statistical Modeling, Linear Algebra, and Discrete Math

RELEVANT TECHNICAL SKILLS

Languages: English (Native), Spanish (Basic), C, C++, Python, x86, OCaml, JavaScript, Java, HTML/CSS, Stata, and R

Web & Mobile: Node.js (React, React Native, Redux, OAuth 2.0), JQuery, AJAX, and Bootstrap **Data Analysis:** ggplot, sklearn, pyplot, numpy, and pandas **Technologies:** Windows, Linux, Excel, GitHub, MongoDB, TensorFlow, and LaTeX

RELEVANT SUMMER EXPERIENCES

Horizons School of Technology *(a MERN stack development boot camp) Fellow Summer 2017*

- Implemented several **MERN** stack application during daily, collaborative coding exercises
- Assisted peers with debugging and conceptual understanding of system level knowledge and algorithms

First National Bank of Omaha *(a midsized commercial bank) Project Management Intern Summer 2015*

- Researched banking preferences of millennials in the area and wrote business case for a product to meet their preferences
- Designed fronted wireframe for a millennial focused mobile banking and financial assistant app

TECH PROJECTS

Campfire *Technology used: Node.js (Hapi, pg, Pug), React, Redux, Firebase and Heroku Winter 2016 - present*

- Utilized **Hapi**, **Heroku**, and **Postgres** to develop a web app to centralize information about local events
- Created fronted initially using **Google Map's Javascript API** and **Pug** templates
- Increased app performance by converting the app to a single page app using **React**, **Redux**, and **Firebase**
- Designed and integrated REST API for IOS app written in **Swift**

Live Class *Technology used: Express, Heroku, React, and Redux Summer 2017*

- Created livestreaming classroom MERN stack web app
- Implemented livestreaming feature using **YouTube's Livestreaming API**

Rattlesnake *Technology used: OCaml and C++ Spring 2017*

- Built a computer for a functional language that supports three primitive value types, first-class functions, TCO, and closures
- Used **C++** to handle garbage collection, system calls, and linkage

RECREATIONAL ACTIVITIES

Co-president, *Swarthmore African American Association*

- Elected to represent African American students to Swarthmore's students and staff
- Organized multiple school wide events and weekly meetings, increasing the number of attendees from the previous year **Tech Chair**, *National Society of Black Engineers*, helped found Swarthmore's chapter, fundraised 10k for trip to national convention

Member, *ABLLE*, mentored high school students and college underclassmen, lead and hosted fundraisers and other large events

Interests: mixed martial arts- American boxing and wing chun; dance- unfundalai and modern