Updated 04/05/22. Curated from <https://www.reddit.com/r/cscareerquestions/comments/iucp54/a_rcscareerquestions_college_survival_guide/>

*Ultimate College Survival Guide*

* Freshman Year
  + Not take 3-4 classes in first semester/quarter
  + Go to professional, social events, make friends
  + Join clubs ([ACM](https://www.acm.org), [WICS](https://web.cs.wpi.edu/~wics/)) & hackathons
  + Try interning in above organizatiosn & events
  + Look for freshman internship (IT deparment, research with professors)
  + Training programs ([Google STEP](https://buildyourfuture.withgoogle.com/programs/step/), [Microsoft Explore](https://careers.microsoft.com/us/en/job/870951/Internship-Opportunities-for-Students-Explore-Program), [Amazon Future Engineer](https://www.amazonfutureengineer.com/), [Uber STARInternhip](https://www.uber.com/global/en/careers/list/62885/), [Facebook University](https://www.facebook.com/careers/students-and-grads/students))
  + Alternative 1: Research smaller companies, reach out to CEO via LinkedIn
  + Alternative 2: University recruiting websites ([Handshake](https://wpi.joinhandshake.com/login))
* Sophomore Year
  + With basic coding knowledge, create own website/GitHub & start contributing small personal projects. Place emphasis on clean code, learning new frameworks & creating robust features
  + In case of no freshman internship/research opportunity, focus on community involvement, projects & hackathons
  + Focus on DS & A course this year
* Junior Year
  + Companies are more willing to hire, as they are likely to convert you into fulltime. Much easier than interviewing & higher TC package
  + Look into coding interview prep sites ([LeetCode](https://leetcode.com), [HackerRank](https://www.hackerrank.com))
  + Read prep books ([CTCI](https://thuvienpdf.com/cracking-the-coding-interview), [EPI](https://elementsofprogramminginterviews.com))
  + Continue hackathons, club/orgs, portfolio
  + Since harder classes, take no more than 2-3 upper division CS courses & 1-2 GE classes. Don’t take more than 16 units
  + Research fields in CS, look up job postings & required skill set on [LinkedIn](https://www.linkedin.com) & [Roadmap.sh](https://roadmap.sh). Can make up for missing skills on [Udemy](https://www.udemy.com)
* Senior Year
  + Last change to improve resume. Focus heavily on senior capstone, project classes, etc.
  + Optional: should have at least 3 GitHub projects to share with recruiter
  + If you get a return offer, don’t immediately sign the offer. Keep interviewing until the offer acceptance deadline.
  + Ask for referrals & have someone to send your resume to recruiters
* Classes
  + Required: Programming in Java/C++/Python (OOP), Boolean Algebra/Discrete Math
  + Must take: Computer Networks, Operating Systems/UNIX, Databases/SQL, Programming Languages,
  + Good to have: Full-stack web development (preferably MERN stack), very deep understanding of at least one language, Semester/Year-Long Capstone course (preferably working with a company), Compilers, Human Computer Interaction
* How to Succeed in Classes
  + [Read before classes](https://m.youtube.com/watch?v=mlk0NAA3I48)
  + [Effective study](https://m.youtube.com/watch?v=Lw4dMehQkgs)
  + Attend every lecture attentively, sit at class front, highlight important information, take notes of questions you might have & possible test questions
  + Compile highlighted information into a study guide
  + Go to office hours & become close with 1-2 professors
  + Make friends in classes
  + Start early on assignments
* Projects
  + Super important, if you lack work experience
  + Edit “Project” section in resume with projects & tech relevant to desired position
  + Start with the end product in mind & draw out the intended architecture/functionalities
* How to Succeed in Class Projects
  + Read requirements, plan your work, draw pictures & ask for help
  + Learn to debug, [clean code](https://medium.com/mindorks/how-to-write-clean-code-lessons-learnt-from-the-clean-code-robert-c-martin-9ffc7aef870c)
* Open Source Contributions
  + Contributions to popular repositories look good on resume
* Hackathons
  + Gain industry knowledge, network with professionals, hone coding skills, learn about companies, attend workshops, ask questions to mentors, get free goodies & applications to internships & full-time positions
  + Awards are resume boosters and talking points during interviews
  + [Major League Hacking](https://mlh.io/)
  + Judges care about the pitch & the idea rather than the execution
* Friends & Networking
  + Career group & non-career group
  + Meets friends through orientation, courses, clubs
* Resume
  + [Write accomplishment as bullet points](https://www.linkedin.com/pulse/20140929001534-24454816-my-personal-formula-for-a-better-resume): Accomplished X by doing Y as measured by Z
  + Don’t put overdone school projects, except capstone projects
  + No need for graphics, colors, fonts, etc.
  + At least 3 bullet points per work experience/project
  + Keep 3.5+ GPA, don’t keep 3.0-
  + Focus on individual contributions & leadership, not the team
  + Sections: Name, School, GPA, Work Experience, Projects, Skills (seperate languages & tools/platforms), relevant links (GitHub, website, portfolio)
  + Get your resume reviewed ([csMajors](https://www.reddit.com/r/csMajors/) & [cscareerquestions](https://www.reddit.com/r/cscareerquestions/) subreddits have weekly resume review thread)
* [Searching for Internships](https://docs.google.com/document/d/1YVi070wa5xPghIlDi5xV7k4FadJV3XYGCVynoFP8dWU/edit)
* [Searching for Full Time Positions](https://docs.google.com/document/d/1u-EyTVmPUFzDB9XeJjnwFeLm0fAVr8QI8OT3sA6Oe_8/edit)
* Final Thoughts