Interview Prep

☑ Checklist

Edwin

A little about me

Full Stack Developer, interested in IOT, robotics and tinkering with consumer electronics.

Working at Browserstack with Ruby on Rails, React and NodeJS.

Robocon Alumni and SIH Mentor. 2019 passout.

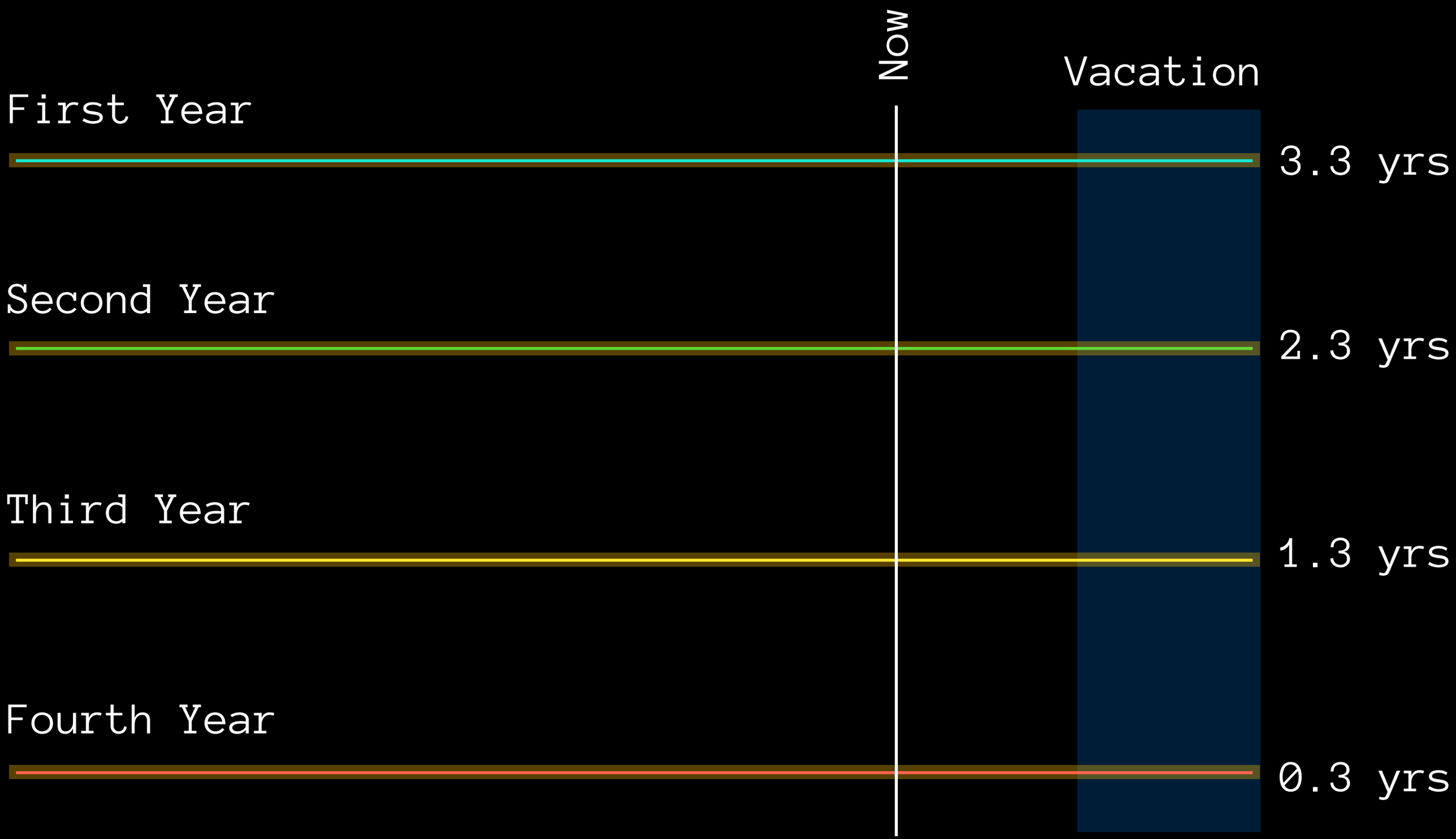
Targeted Audience

This talk is focused at those who aim to be software developers as my experience is only suited for this specific domain.

I am focusing on how to prepare yourself for an interview, specifically, your technical skills.

Other vital skills necessary(like soft skills, competitive coding, etc) will be taken up by other speakers more skilled/suited in those fields.

Your Current Scenario



Rough Outline

First Year

Second Year

Third Year

Fourth Year

Foundational Years

Learn DSA/

Learn some lang/ framework

Set some companies as target

First Year

- Go back and refresh on C. No need to master it. But understand the basics of coding(Variables, Data types, Arrays, Looping, Conditionals, Functions)
- If you like challenges, go and check out https://www.hackerrank.com/domains/cpp for some nice intro to object oriented programming(C++). Java also works.
- Interact with seniors about their projects and extra curricular activities.
- Enjoy College life, best 4 years for a while. (*circa.* before coronavirus)

Internships

- **It is very crucial**, enough said.
- Internships give exposure. It shows that DSA is important to get the job. But skills required in jobs rank like this
  - Communication skills
  - Knowledge
  - Team work
- Internshala/Frapp are a good place to get internships. LinkedIn has a section for internships too. Alumni can also get you some. Ask the student alumni head to get in touch
- NGOs(TeachForIndia) also do give tech internships. They also give brownie points for Social Help.

Second Year

- Learn one language that has Classes (Java, C++, C#, Python, Ruby, Swift, NOT Javascript). You should be clear on this and not need IDE(autocorrect) to implement code on this. Sublime Text is a good editor

- Go online, learn about algorithms, beyond the college syllabus (Graphs, Red-Black Trees, AVL Tree, B-Trees, Dynamic Algorithms). GeeksForGeeks, LeetCode, HackerEarth, HackerRank, Coursera are good resources.

- Start a couple of small projects(Go beyond what the curriculum demands). These should be deployed on some online server(Start with Heroku or Netlify. Go for AWS/DigitalOcean for Pro-level stuff.)

- Start going to Hackathons. Look at others solutions and learn what they do. Expand your connections, someday you might get a job from them.

Second Year(Cont.)

- Start working on Competitive Coding.
- Attend Conferences, JSConf happens in Mumbai every quarter, PyConf happens across India. Many others happen in Mumbai. These give excellent industry exposure and the trends.
- Look into certifying yourself. OCPJP is one such for that Java. Cloud is on the rise, so that’s another avenue. Coursera/Udacity are good places to learn. Prof. Mahendra Mehra has a lot of certifications, you can reach out to him if you have doubts on what you need to do.
- Start researching about Dream Companies(Google, Facebook, Amazon), preparation is the key. One of you is gonna get it eventually. It’s up to you to grab the opportunity.

Third Year

- You choose a target company(or stack). Ask alumni/ professors for help on this. Start Building projects on this. A Todo app is a great start. Use Git(otherwise you will **** up someday and loose your work). Use DevOps to deploy this(Docker for Sandboxing/Repeatability, Travis CI for deploys, Selenium for testing). Get familiar with Unit Testing.

- Start maintaining Github Profile and add more projects.

- Get familiar with Linux, switch your daily machine to it. Hard in the beginning, but this opens up a ton of job opportunities.

Third Year(Cont.)

- Web is the current trend. So be clear on how HTTP works. Request/Response paradigm should be clear.
- Also learn some other language(stack) so that you have experience in what are the similarities/ differences. This helps in understanding the critical parts for a framework.
- Research your targeted company and learn what problems they solve. This helps a lot in interviews(When they ask you to ask questions)
- Make a schedule, learn how to get past the Aptitude test. Go through books and make sure that you can at least get to the interview. Otherwise all the prior work is pointless.

Just before Interviews

- Be Confident in the interview. Fake it till you make it.
- Check out online resources(GeekForGeeks/GlassDoor) to get the common interview questions for the specific company.
- Go through Riddles/Puzzles, some companies really love to ask them in interview.
- Prep with your friends to build confidence. Sometimes all you need is a couple of trials with friends to make the interviewing skills perfect.
- A bit controversial point, but try to be one of the first 10 ppl to give the interview. The interviewer gets bored after the first few ppl and starts comparing.

Fourth Year

- Well, unfortunately this is a bad time to search for a job.
- Fortunately, You want a CS job. That means remote work is possible.
- The Best approach now is
- Prep up skills in the lot of free time you now have.
- Try getting freelance jobs online(Freelancers/ Upwork/ Toptal)
- Look for remote job(cutshort.io, Stackoverflow, Linkedin(as a LinkedIn Post not bio))
- Pray, Godspeed.

Credits

To these fine friends of mine who helped me compile this list. Special regards to Shweta who stayed up late to make this ppt.

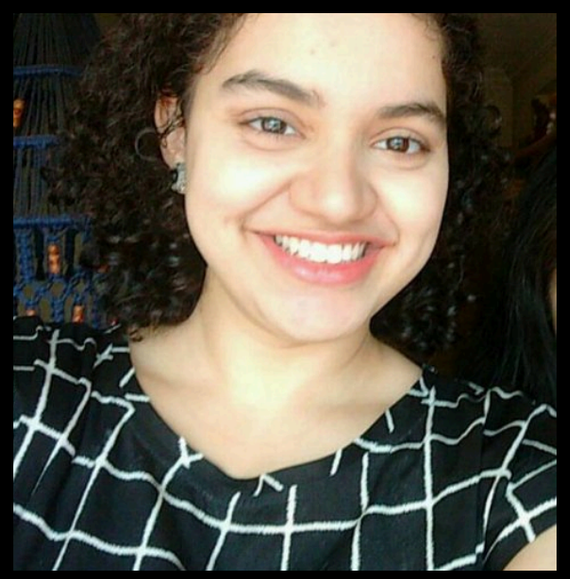
Bhanu

Kaumudi

Uday

Joshua

Samson



Shweta

FAQ