BETA
News from Computer Science and Engineering

Hackathon Edition

13 September 2018,
Semester 2

ISSUE 140
We then round off this issue with an interview with Renee Noble from the Girls’ Programming Network. So huge thanks to Renee and the hackathon interviewees (Jason Quek and Kevin Ma) for taking out the time to speak to us!

I highly encourage you to sign up for the hackathon here. It’s a great learning experience and one you’ll never forget.

In other news, we recently conducted our AGM (Annual General Meeting) for this year and nominations for exec positions are now open! You can find all the info you need here. Feel free to email us at info@csesoc.org.au if you need more insight into the exec/director roles!

We hope you have a great midsem break and look forward to seeing you at our hackathon next week!

Siddhant Virmani
Editor in Chief | Publications Director 2018
What is a HACKATHON?
Adrian Martinez

With CSESoc’s very own hackathon right around the corner, you might be wondering what a hackathon even is. Adrian Martinez writes about his first experience at a hackathon and why you should be excited to attend CSESoc’s!

Computer Science student Yang Sim just wanted to sleep. He’s hunched over his folded arms, glancing over lines and lines of documentation that, he hopes, will put a couple of buttons and a map on the app. Next to him, you couldn’t see much of Josie Anugerah beneath a pawprint face mask and sith-lord hoodie, but you could just tell she was this close to letting loose a manic laugh of cathartic release from the stress. “I’m gonna get a bit of rest, see y’all in the morning,” she says, adding one more to the dozen of empty seats around tables, to join the horde of hackers passed out on anything that could be considered flat - floors, tables, under tables, hugging the office plant pots - being picky never got anyone anywhere.

It’s around 4 a.m at Facebook Sydney offices, about 8 hours into the annual Facebook Sydney Hackathon and despite the bags under eyes, grimace at compile errors, or even worse, shrugging off a successful compiler that definitely shouldn’t have worked, there was a real sense of wonder and comradery that you can’t get from smashing your homework assignments. Across from Yang, Sebastian Chua sculled down a red bull in one hand, then a sugar-free kombucha in the other - if it wasn’t for the Facebook swag, uni students glued to their screens, and arrays of tables, you could’ve sworn it was the aftermath of a colossal house party. In other rooms, shots of coffee were used as a measure of how hard you were working at your product, and in some others, it was dead quiet - headphones in, concentration on.

When you think of an overnight hackathon it’s more like this, and less of the stuff in the Social Network where college kids take shots per ten lines of code, a shot every three minutes, and a shot when they reach checkpoints (how on Earth would you survive 24 hours like that!?). In the CSESoc Hackathon, you’ll be around 150 passionate people, who are excited to get free swag and food and flesh out an idea in the twelve hours you have. Whether you’re a fresher who’s just finished COMP1511 or an experienced dev who’s...
clawed through the deep night trenches of AOS both of you have just about the same chance of winning as the other.

A lot hangs on the pitch. Even if your code is perfect but the compiler is broken and you had to make a powerpoint mock app - if your presentation, idea and visible functionality is top notch then you’ll still be alright. With that said, it’s definitely less about winning, and more about what you and your team are going to learn while building amongst a ton of caffeine riddled students and experienced mentors.

But Adrian! I’m not about that ultra-competitive, build things that disrupt industries (classic) fiasco! Hey, same here. Instead, let’s just all build emoji renderers that can handle hundreds of thousands emojis at once at playable frames per second! Or mingle the two ultimate productivity suites in Trello and Google Calendar and bind them together to finally produce the best task manager (please, this would be great). Instead of using hackathons as an ultra competitive vent - which you can do, no one is stopping you - you can go into it expecting to learn as much as you can, eat tons of delicious food, made even more scrumptious with the fact that it's free, and hack shit with a bunch of soon-to-be best mates.

You never know what will happen when you step into a hackathon and hey, that might be the best bit.

CSE REVUE 2018: Mycrosoftr Shrexcel

Somebody once told me that CSE Revue: Mycrosoftr Shrexcel is coming to a Science Theatre not so far far away. CSE Revue is a student run sketch comedy show with lots of layers, including acting, dancing, singing, videos and even a live band! It is entirely written, directed, produced and performed by (mostly) CSE students! So come to shrexperience a shrexciting night of shrexceptional entertainment that will definitely shrexceed your shrexpectations!
If you are worried whether if you're skilled enough at coding to attend, have a look at what James Pan, a coding newbie, has to say about attending this hackathon!

What were your initial impressions of a hackathon?

My initial impressions of what a hackathon was that it was a 'hacking' competition where you attempt to hack into a website and whoever gets the furthest, wins the competition. Clearly, as many of you probably already know, that impression was wrong. When I talked to more experienced coders that have participated in hackathons, I slowly began to realise that hackathons are more similar to normal coding competitions where you group up with other people and attempt to create an application together that hopefully might solve one of the many problems that are relevant today.

Why have you decided to participate in this hackathon?

Participating in this hackathon is a rare opportunity to code together in a group. Usually in uni courses contain mostly of individual coding projects rather than team-oriented ones, and that doesn’t really match what the the big technology companies actually do on the day-to-day basis. The hackathon also acts as somewhat of an incubator for raw technical skills, as it forces you to learn a lot of new ways to solve issues behind applications in a short amount of time.
What do you hope to achieve from this hackathon?
Personally, I don't have much experience in team coding, especially in a group setting trying to create an application together. This process of teamwork in coding in general is a key skill that is required in a lot of tech jobs and a lot of big companies would rather you work together in a team, review other people's code and figure out ways to improve it. The hackathon is a prime example of how this process goes, and is definitely an excellent place to familiarise yourself with team projects.

Any words for those still tentative on whether they should join the hackathon?
The hackathon is a very valuable way to learn a lot of new skill sets in such a short time due to the nature of the competition, and can also help you develop new aspects of your knowledge that you haven't necessarily explored before. Because of this, I would highly encourage those with any slight interest to participate, there's no reason not to!

Hackathons aren't just for developers, Jason Quek is a Biomechatronic Engineering student and below are his impressions of how important hackathons are in developing yourself.

What were your initial impressions of a hackathon and has your perception of hackathons changed after participating in one?
Initial impressions were of the more traditional ones where people code for 48+ hrs straight with no sleep. Of course, my first hackathon demonstrated that they can move beyond coding and focus on fleshing out and refining raw ideas.

What was a highlight from your first hackathon?
Being able to work with people who aren’t necessarily undergrads (my first hackathon team mate was semi-retired).

You definitely learn more about different ideation and execution methodologies due to the greater depth and breadth of experience.

Did you gain anything valuable from your first hackathon? E.g. network, new skill, self-confidence etc.
My first hackathon was my first exposure to the entrepreneurial mindset along with the pitching and Q&A format. Along with piquing my interest in entrepreneurship, it also helped me discover my pitching and Q&A capabilities whilst teaching me how to keep calm and improvise under pressure.

Any advice for people about to attend their first hackathon?
Don’t do your first hackathon with your friends. While working with similar-minded people is appealing, you won’t necessarily get much exposure to new thinking processes and/or knowledge, or learn how to work with different people. Hackathons are short and sweet with opportunities for networking, so they present a great opportunity for you to try new things and form new connections.

Finally, for those looking for an expert insight into how hackathons run, continue below to have a read of what Kevin Ma has to say about his experience with hackathons.

Past hackathon achievements?
- CSE Soc Hackathon 2016 Matchr (Tinder for developers), 3rd place
- StartCon Hackathon 2016 BACTracker (BAC monitoring), Honourable Mention
- StartCon Hackathon 2017 GoHackUs (crowdsourced bug bounty platform), 2nd place

What were your initial impressions after your first hackathon?
My first hackathon was the 2016 CSE Soc hackathon. It was surprisingly, a fairly low-key event with a modest turnout of five teams. This was probably a blessing since I had no idea what I was getting myself into - I
was half expecting an intense faceoff, fingers in a furious flurry of typing all day. Of course, the reality was far from the case, and while everyone was serious about winning, no one could work so long without a break, or without generously provided sustenance in the form of snacks and lunch.

Over the two days it ran, there were definitely ups and downs - we made steady progress, but I was particularly struggling to make meaningful contributions. Time drained away as I futilely read docs and consulted StackOverflow to get an authentication system working. I realised far too late that there was no real point even getting it working, since we could just hard-code the login anyway. However, thanks to the valiant efforts of my teammates, we pulled through and had a functioning product to show. Just the opportunity to build something cool with friends and experiment with a bunch of technologies I had never touched was a great reward in itself.

What do you think you can someone can gain out of attending a hackathon?
Friends, fun, and future job prospects! Okay, it's never that easy, but I can honestly say that hackathons are great for learning to work fast and picking things up on the fly. They also represent an opportunity to flex your creativity and build things that would never come up during the course of normal study. Many hackathons are organised and sponsored by software companies, giving a chance to chat with industry professionals. If they happen to be the judges and you impress them, even better!

Do you have tips for people attending?
To get the most out of your time, use existing libraries and tooling from the open source community! As a frontend developer, often most hackathon projects boil down to making a simple web application, so use whatever frontend framework you or your team are most familiar with. From my experience, one of the main hurdles is bootstrapping the project, which can be a massive hassle and time sink if not accounted for. To get the smoothest and fastest setup time, I would recommend using a CLI (command line interface) option for any one of the popular frameworks, such as:

- **create-react-app**
  https://github.com/facebook/create-react-app

- **vue-cli**
  https://github.com/vuejs/vue-cli

- **angular-cli**
  https://github.com/angular/angular-cli

There's no shortage of great, free resources for web development, it's just a matter of taking some time to learn them so you're ready to go on the day of the hackathon. For those who are already well-versed in web development, linters such as ESLint (https://eslint.org) and TSLint (https://palantir.github.io/tslint) help avoid common runtime errors, formatters such as Prettier (https://prettier.io) shave off formatting time, and proper editor integration (shoutout to Visual Studio Code) can really accelerate the coding process.

Another key aspect of hackathons is the pitch. This is the main opportunity for your team to show off the amazing product you have built, and convince the judges your team is worthy of the prize. If you've read any article related to hackathons or attended one, it's almost a meme how often this point is reiterated. But here I will say it again, because it really does make your team stand out - try to deliver an excellent pitch and an even better demo. Part of this is down to communication, and the other part to content, but make sure your team starts thinking about the pitch before building anything. It's difficult to justify working on a feature that's not part of the core functionality, or won't sell your team's idea better, given the limited time.

For something more concrete, try performing research around the problem your product tries to solve. Why does this problem still exist, and how does the product address it? What are similar, competing products doing, and how is yours different? Data and numbers are your friend here, so don't hesitate to stick in graphs or bold figures on your slides to lend credibility to your argument - just ensure they are relevant to the discussion.
On the topic of slides, it is best to avoid making the content overly verbose by jamming blocks of text into one slide. Each slide should help focus the attention of the audience on one main point, not distract them with dense arrays of images or text. Small details such as consistency of formatting and noteless delivery give off a sense of polish and professionalism that can only come with investing time into slide preparation and rehearsal.

In terms of the demo, it is only necessary to create the minimum working product that captures the main idea. This means your team should ideally prioritise building and refining features that make up this main idea, before doing anything else. This is difficult since there are usually several people working at once, so it can be helpful to split team members by role, e.g. two frontend and two backend. In a hypothetical project to make a Tinder clone, this might mean two people creating the core user flow from home page to signup, login, browse and chat, and the other two wiring up database queries and API calls. That being said, remember that anything can be mocked or hard-coded in case of time constraints.

Having said all this, I still think the most important thing is to approach hackathons with the mentality to learn. To me, it was a new way to approach programming, one that didn’t care about future-proofing or ‘good’ design, but rather a lesson in working fast, being decisive and winging it. Whether it’s figuring out how a library works, making that hero image line up perfectly, or writing your first pitch, there’s something to gain for everyone. So regardless of your background, I’d highly encourage trying hackathons out - with just a couple of mates and an idea, you can be certain of an experience worth your while.

How do you prepare for one? Is it possible to?
As the saying goes, the early bird catches the worm, and starting preparation early pays off exponentially later. However, as its name suggests, most of the hacking should be done when the event begins, not beforehand - judges can usually tell between something coded up live and a ready-made product. As mentioned, the preparation should mostly involve making your team familiar with the technology stack being used, and bootstrapping the project so that you can avoid writing boilerplate code and jump straight in. In general, anything that will speed up the time to writing the first page, such as brainstorming ideas ahead of time and setting up development environments, will be useful.

So what are you waiting for? Grab your laptop, charger, headphones, and coffee for a hackathon to remember!
Not long ago I came across a Facebook post published by the Girl's Programming Network (GPN) asking for volunteers who could help teach girls how to code. GPN is a community of women and girls who have a similar interest in computer science and is a network that I wish I knew more about when I was younger. I decided to sign up to the network and was on my way to a "Hack-tivity" Hustle where all of the tutors came together over a weekend to learn new things, make friends, and create coding and unplugged computer science activities for kids. I left that event feeling more inspired than ever to help teach and encourage girls to code. The community was warm, welcoming and unique and so, to learn more about this network, I met with the Director of GPN Australia, Renee Noble, to learn more:

Firstly, how did the GPN get started?
It started 10 years ago, it's our 10th birthday this year which is exciting! There were about 5 women involved at the start and they wanted to just do something for girls and it came off the idea of 'we will teach girls and we will get better at coding while we do it.'

Was everyone at the start from a computer science field?
They all wondered their way into coding from various other things. Some of them had done arts degrees and then came back to university to do an IT masters afterwards. So many had to take a strange path to get here and we want to make sure girls can get here without having to take a strange path.

So is that why you think high school students to start learning to code now?
Yeah, it's a great skill to get your head around earlier. We start teaching maths in kindergarten so why would you start teaching coding at 18? It's just another skill or way of breaking down problems. It's a good skill to have in general if you're going to be a software engineer or not. Or you might use coding as a tool to do any kind of work really. And we want to teach it to kids in high school because that's the only chance we get to teach it to everybody, and if we just wait for people to self-select into something, people are going to miss the chance to learn about coding.

And is that why the network focuses just on girls?
Yes, well it comes from the huge underrepresentation of women in Tech. At universities I think it's only 15% of IT students are females. That's one of the reasons, and it's not just about the girls that we teach but also the women who get involved as well. So they get involved, they teach, they get better at coding, then more confident in their own skills, then leadership opportunities, and just become super awesome people who are confident and competent. And coming out of GPN, you make new friends and have mentors you can go to. It's not about making the entry into computer science bigger for girls, but making sure they stay in all the way along.
How did you first get involved with Coding?
When I entered university I didn’t know anything about coding, I was studying Chemical Engineering. But one night a bunch of my friends that I lived with in college convinced me to learn how to code. I thought it was crazy I didn’t know you could just DO programming on a regular computer I thought you needed a PhD or something. I had no concept of what it was, or just never thought about it. I was then convinced to do a subject and I kind of rolled from there and realised I’m really good at algorithms and how you could do much stuff with coding. I combined it with chemical engineering and while people were doing their honours thesis and waiting for paint to dry I was just writing code and then I hit run and it would run a thousand experiments in a few seconds. I realised the power of it and how you could do so much more.

How has joining the network helped you?
I had imposter syndrome and so one of my friends convinced me that I should do some tutoring at GPN. He knew the people who started it. I signed up and got involved. I realised soon after that not everyone was a super nerd and I related much more to the girls at GPN. I was doing a lecture on dictionaries one day and the slides didn’t work at the start so I was writing on the whiteboard, then halfway through they came back up and everyone was helping me to get back to the right slides. It’s so supportive. The tutors are nice, the girls are nice and they just want you to do your best.

What is one your fondest memories from GPN?
So many but this one day in particular a girl came in with her mum and she had anxiety issues and couldn’t really handle being there. And she was running off and crying and her mum was trying really hard to get her inside saying, ‘she’s really good. I really want her to come because she’s made these websites, and she’s really into Minecraft and builds all these things and I think it would be good for her.’ And so mum is really upset because she just wants her daughter to get the best experience and she knows she’ll have a good time if she could get her to come along. So we get her to come in and we give her a buddy and eventually she settles in to the room. And then we get split up to start doing the coding activities and the next time I see her it’s the end of the day and I see her with her arms around two people who are her new best friends and she’s talking about coming back and how she doesn’t want to go home. So yeah GPN helps you overcome all kinds of things, and everyone can find something at GPN.

Each term they run a workshop where girls can learn about an exciting topic in computer science, ranging from cellular automata to artificial intelligence. In most workshops they look at python, a programming language, to gain hands-on experience with writing computer programs that do what you want.

Special thank you to Renee Noble for taking the time for this interview!