Fresh out of high school and into the new world of university. The nervousness and excitement was overwhelming, especially since this was my first time moving out of the comfort of home, much less moving to a new country. The extreme pace of university life and studies was, however, quick to get me out of my homesickness and fear of the unknown as I delved into the subjects of year 1 at NTU. A year full of “firsts”, my first year at university was definitely a one to remember for a long time!

**NTU Year 1**

ENGINEERING MATH

The first part of Engineering Math 1 was linear algebra, so it dealt with matrices, n-dimensional vectors, linear equations and the like. I was able to get through these chapters using class notes from lectures supported by some of my high school knowledge. But the second half touched on one of my weakest points - statistics. I had always been weak at statistics because of its heavy reliance on formulas and mechanical calculations.

However, at the end of the course, I was able to develop some knowledge in statistics which proved to be fairly useful in my year 2 research project on deep learning.

Engineering Math 2 was the complete opposite of its concurrent course Engineering Math 1. Math 2 was my absolute favorite course in the whole first year, mainly because I was (and still am) a huge fan of calculus. I loved integration and differentiation which formed the bulk of this course. I was able to breeze through the first half fairly easily using my high school knowledge but the second half was where I hit somewhat of a stumbling block. This is when they first introduced sequences and series, a topic I had only ever briefly encountered once before during my AP exam preparations. At this point in university, I had fallen under the impression that without my high school knowledge I would not be able to manage the courses. After a short period of worry and stress, I was able to find some resources on the internet to supplement the lecture notes from university. This also happened to be my first time studying something from the internet without a “syllabus” to guide me (one of the many lessons I learnt during my time at university - there is no such thing as a “syllabus” after school, you just make do with whatever you can find). Those notes (and of course my efforts) were what got me through Math 2. Overall Engineering Math 2 was a great course which allowed me to explore my interest in calculus, which later helped me decide to specialize in Artificial Intelligence.

DIGITAL LOGIC

Digital Logic was another instance of something I had almost never seen before. It was an introduction to digital circuit design and my first ever exposure to hardware as a subject. This one started out simple and taught things like gate level design and truth tables; later, it delved into synchronous circuit elements and Verilog programming. This was one of the more challenging courses of the first semester, and the fact that I had no previous experience with it, worried me more. However as the course progressed, I was able to grasp the concepts and developed an interest in them which further helped me crystallize my understanding. Another skill I was able to pick up through this course was Verilog programming because it closely resembles C programming. I add a special mention to the professors who taught this course as their teaching was the reason I was able to gain more knowledge on hardware.

ICT

I had already had many “first” experiences in university, and adding to the list is one more from the course titled Introduction to Computational Thinking (ICT). It was learning Python programming. I had never seen Python code in my life before so I was all the more shocked when I heard that this whole course would be based on Python. The labs were all based on the Raspberry Pi board (whose packaging box I was very fond of back home because the logo is very cute, but that was pretty much all I knew about it). I had an especially hard time adjusting to Python syntax since it’s so verbal and English-like, but as the labs progressed I started getting more comfortable with it. In the final lab, I was able to write the Python code to model a rolling ball game on the RPi SenseHat display module. My experiences with Google from Engineering Math 2 encouraged me to use the internet without being afraid of finding complicated, confidence destroying answers to my basic questions.

The course also involved a (very large) “mini project”. It was to construct a full fledged canteen recommendation system with UI graphics using Python. The final few weeks were dedicated to the project and contributed significantly to improving my comfort with Python, which was at first unusual and unfamiliar. The project exposed me to the more nuanced aspects of a new programming language, the greatest example being industry standard syntax and best practices. All in all, as challenging as this course was, it was the one that benefited me the most in my following years at university as Python formed an integral part of another course in the next semester as well as my year 2 research project.

Another new thing this course taught was data structures in Python. Though they only taught only binary trees and nothing more, I found myself lost throughout that section and all the Python codes, which I just had started feeling comfortable with, started seeming like Latin and Greek again. However after some intense research effort, I was able to understand those concepts.

My main takeaway from this course, more than all the Python, was perseverance. There were many points during this course where I was demoralized but I kept telling myself not to give up and keep trying and the results were fantastic! I scored well in ICT and am now more comfortable with Python than ever(though this is supported by my future endeavors in Python as well) ICT formed a strong base for me to learn Python as well as provided me with an opportunity to become mentally stronger.

DISCRETE MATH

The second round of math at NTU was Discrete Math. This module was more about logical operations than calculations. Some of the topics taught in this module were logical and, or, not, xor operations and solving expressions using their properties; truth tables, recurrence equations and graphs. They also touched a bit on Euler paths/circuits and Hamilton paths/circuits. These were slightly more complex concepts than the others but were interesting nonetheless!

This course was one of the easier modules as it quite simple to pick up and understand the concepts. But the downside of this mod was that there was a lot of scope for twisting the questions. The questions were usually worded differently from the tutorials and tested multiple concepts together to increase the complexity. The final exam of Discrete Math was the epitome of this as the questions were deceptively simple and demanded deeper knowledge of the concepts.

This is something I had not realized until I started solving the past exam papers for this subject. The questions seemed so easy at first glance but when I got to solving them they were much more than that. I realized the burning need to further understand the concept than just the formulas given in the notes. I needed to solve more “application-based” questions to support the direct ones in the tutorials. This led to more internet research and reading on the topics until I was in a position to be able to solve the even the twisted questions.

From this module I was able to appreciate the need for attention to detail and how important it is to go beyond superficial learning.

COA

The second interaction I had with hardware at NTU was through Computer Organization and Architecture. It introduced concepts on instruction sets, computer arithmetic, CPU/ALU, pipelining and data-transfer. This was the most difficult course for me during my second semester at NTU. Despite the first half being at least doable, the second half was a lot more complex and demanding in terms of understanding as well as practice.

Data-transfer had me worried for a large part of the semester. I found it very difficult to fathom how two devices could “agree” on a transfer rate before starting data transfer between them and not messing anything up. UART was one of the most complex and insane topics for me because none of it made sense! The tutorials were of some help but I found them insufficient and the exam questions were beyond my understanding at that point, which is why I told myself I would attempt them later after I get some concepts cleared. However, till the day before the exam I was not clear on the concept and so unable to solve the questions. I was also worried that the exam will be filled with only questions from this topic.

So I spoke to some people for advice, and they told me to complete the topics I was confident in and just brush through the rest. I followed that, but the problem was that I was confident in probably 3-4 out of around 10 topics, which had me really worried. However, it was already too late then so I decided to just go write whatever I knew. They say luck comes in unexpected places, and this exam was one of them for me. I was so worried that the paper would be filler with UART and not one question came from there! Maybe the examiners felt that many of the students would have difficulty with that topic so they decided to have some mercy on us :) Whatever be the case, even though they never tested us on that topic, I was not well prepared with the other topics either. Considering that, I was proud that I scored above a B- grade.

One of the most important lessons I learnt from my whole first year is from this module: it’s okay to not be 100% prepared for an exam! I learnt that not every module will be doable and some of them won’t make sense to me till the end and that’s fine! What matters is the effort and hard work that goes into preparing.

DATA STRUCTURES

Data structures was my solace among all the other hectic subjects of semester 2. I loved C programming since before university and the course was also taught in a very hands-on way with more emphasis on labs than the lectures and theory. Another plus point for this course is the fact that it dealt with coding(and I love coding).

The start of the course was like high school revision where they taught things like basic C syntax, conditional constructs, loops, function defining and recursion. They had weekly assignments and test submissions which I always did in advance because I enjoyed it so much(and as a way to put off doing other subjects, but no one needs to know that). i was quite familiar with the topics in the first half which added to my enjoyment of the course!

The second half challenged my learning capacity as they introduced more complex data structures such as linked lists, stacks and queues as well as an introduction to binary trees and binary search trees(BST) in C. But these topics interested me so I was able to pick them up fast. Also having a base introduction from ICT in Python also helped understand some of the intuition behind the slightly more challenging BT/BST better.

The learning experience in this course was definitely more in the second half, supported by the first half which served as a good refresher. Nothing much happened apart from that but this course is close to my heart for two reasons, first because it was my first so-called “easy-A” in NTU; and the second, more important reason, is that this course made me feel a little less scared of university in general. I was still sort of adjusting to university after my first December break and dealing with homesickness, but doing coding made me forget all of that and focus on the joy of making a program work!

DSAI

Talking robots are a thing of the past. Today, we want self-learning robots who can both be our calculator and our friend. The course Data Science and Artificial Intelligence showed us how to take the first baby step into the new and upcoming field. So my batch was the first to experience this course and it was one of a kind!

The beginnings of this course were confusing and hazy where I felt like I was being blindly led through a dark tunnel. At one point I was collecting and cleaning data and at the next I was interpreting a regression graph, I wasn’t able to connect the dots until a very informative meeting with the professor taking the subject. He showed me YouTube videos on the advancements of AI to give me a bigger picture and then narrowed it down to show me where I stand in that picture. He explained the intention behind each topic taught and how they would eventually connect together to form a part of the big picture. That 45-minute long meeting helped me piece together the concepts very clearly and intuitively.

The course was conducted completely in Python, something I had experience with from the previous semester. So I was able to manage the coding part after some practice as well.

There was another very large “mini-project” in this course as well(just like in ICT). Only this time, we had to do more than just recommend canteens, we had to do something that could really be used in reality. The problem? We didn’t have a problem statement. The professor handed us a website link and 3 datasets to choose from, and asked us to make our own question. Also the groups were also randomly allocated to us, which meant that I would have to work with people who I had never seen before and could potentially be unhelpful freeloaders.

But as luck shines in unexpected places, I was placed in the best team I could ever hope for. My group members were the most proactive and hard-working people I had ever seen in university, and it was a pleasure working with them. It’s only no wonder we ended up taking all future classes together so that we could team up for projects again.

Back to the “large” mini project. The first few weeks of the allocated time of the project went in deciding the question and reviewing it with the professor. After that, we then started on the coding. We decided to do a happiness index prediction model and analyse with interactive graphs. After a lot of hard work, we were able to make the model and get average prediction results as well. Presenting the project was also a good experience for me(I have public speaking fright) but since I was clear on our work and the concepts we used, I was able to present without a hitch.

This course was the major factor why I decided to do my URECA project in year 2 on deep learning. DSAI was a course that opened to me the doors to a whole new world and I’m beyond excited to see what lies in store for me there!

**NTU Year 2**

ALGO

Algorithms was one of the more challenging courses in the second year. It was quite demanding in that the students were required to understand and memorize codes for various algorithms such as searching, sorting and graph algorithms such as Dijkstra’s and Prim’s algorithm. In addition to code, we also had to memorize the complexities in the best, average and worst case and also know how to derive them.

The major problem I faced in this course was understanding the best worst and average case analysis of algorithms. Despite going through various YouTube videos and online resources, I was unable to solve the questions asked in the course. Despite my best efforts and solving all the tutorials and past papers twice before the exam, I could not do it well. But I made sure not to be weighed down by it and worked all the more hard for the other parts of the course such as lab work.

We had lab presentations in groups every two weeks which required extensive coding to solve one of the questions given and present our solution code as well as analysis of running time change with change in input size. We were needed to make a PowerPoint presentation stating our rationale for using the chosen algorithm, how our solution implements it and show a live demo.

This group work is an example of a situation where I was faced with a difficult group member. Without revealing too much, this group member was a final year student who I was hoping to get some advice from regarding course matters, considering I was his junior. However, the situation completely reversed itself as I was put in the position of helping him understand the coding we did for the lab and directing his every move in the project including writing his speaker notes for the presentations. There was also a point where all the group members were looking to me for confirmation and validation of their work which was also very stressful for me to manage. However, despite the extra work, I took it as a learning experience and took up the responsibility as the “unofficial team leader”. In the end we did well in the group presentations and scored an A grade.

One major thing I learnt from my experience is that in group projects, someone always has to go the extra mile to ensure everything is in order and that one member always has to have a hand in all parts of the project. Having personally experienced that, I realized that it is not an easy job but it does have a rewarding satisfaction to know that I am perceived to be responsible enough for the team to put me in that position. This group experience also disillusioned me on the fact that I will not be as lucky as DSAI in every group project. I am now prepared to face more group projects and confident to do better in the future!

OODP

This was the first official Java programming course at NTU. Though the emphasis was on Object Oriented Programming styles, the majority of the course content was taught in Java except one chapter in C++. In the first half, the course covered all of the standard OOP concepts such as Encapsulation, Inheritance, Polymorphism and Abstraction. These topics were similar to what I had learnt in high school so I was able to cope with them easily. The latter half was based more on the UML and representation side; it covered class diagrams, sequence diagrams and UML relationships such as aggregation and composition.

The course also had a final group project, where we had to create a simple movie booking application. The features needed for the project include listing movie screenings, ticket reservation and admin functions such as editing the movie information, changing currently viewing movies and so on. We employed many aspects that we learnt in the course including design principles such as Open-Closed Principle, Liskov Substitution Principle etc. We also employed many professional best practices and error checking mechanisms. This project helped me learn many new things in terms of design principles and writing good quality code. As I was already quite familiar with Java programming, this project served as a good opportunity to improve my skills and also my code styling and quality.

This project also involved a report about the design principles and OOP concepts we used, and the test cases we validated our application on. Writing the report was a good opportunity for me to learn good documentation practices, and learn how best to explain technical aspects on paper. The skills i acquired here were also very helpful in a future documentation-centric project in another course called Software Engineering in the next semester.

OS

Operating Systems was an interesting course in my fourth semester at university. It was slightly challenging towards the end when the list of algorithms we had to keep in mind suddenly became very long, but it was interesting nonetheless. The main reason I liked this course was because the concepts made proper sense to me. But the problem came in application, I was unsure how to use the theory in the practical questions when the exact variables given were not clear to me. In other words, I found it difficult to correlate the questions to the theory.

In particular, I was most confused with the memory paging algorithm where the memory is organized in a hierarchical structure. There were few concrete formulas and too many different types of questions which I found fairly confusing. One of the other topics I was confused in was mutex and semaphores. The questions from this topic were of two types: the first was insert the mutex/semaphore to ensure the satisfaction of some constraints, the second was to check if the given code satisfies some constraints. These questions needed a lot of practice in order to be able to see the different possible scenarios of context switching.

Of course another slightly less significant problem was my careless calculation errors, which I have always had trouble with, but which posed a problem here because of the extensive calculations involved in this subject. For example, we would be asked to calculate the time taken for a a cache cell to be replaced when using the LRU(Least Recently Used) replacement algorithm. Many times I would miss a decimal point or copy the wrong numbers from the question. However, practice made me confident to trust my calculations, but the main problem of application still remained.

I did a lot of research, and re-watched the video lectures in order to try and find out hints on the application of the concepts. The video lectures helped because the professors taught the course very well, but I was still not fully confident. I realized that the problem I faced is in my opinion the worst kind, where the concepts themselves make sense but applying them is difficult. The natural conclusion that followed is that I needed more practice. So I decided to re-solve the tutorials, and this is where I made the most progress. I was able to pick up the hints from the questions and I was able to solve most of them.

Operating Systems also had labs which were quite interesting as they were closely related to the theory, and showed real applications of it. I had fun doing them with my peers, and we used to help each other a lot in the labs. The lab assessment was also quite unique as it was an oral assessment to be done in pairs. Overall the labs were an important aspect of the course and made it more interesting.

For me, this course confirmed that if a topic is interesting, studying it flows naturally. I enjoyed finding out more about the topics I was weak in as I was interested in them.

uP

Microprocessors was the most difficult course in second year. The main concepts were not very heavy, but the labs and tutorials were extremely challenging. It was hard even to understand the tutorial questions, much less solve them. As for labs, the questions were clear but we were given very few hints on how to solve them. However there was some reference code for each lab which was quite helpful.

I did most of the labs beforehand with my peers as we found the stipulated 3 hours insufficient to complete the labs, as did most others. Almost all of my free afternoons were spent doing Microprocessor labs or deciphering the reference codes. Essentially, the labs involved building up the functionality of a robot, until at the final lab it could move in all four directions, use a tachometer to show the distance to a nearby object, use an IR sensor to detect the presence of a line under it and a few more. Each functionality was its own lab. We had to read the lab manual for instructions and reference the sample code to build the actual code needed to implement the function needed. When we had questions, the supervisor would only give us vague theoretical hints which we did not find very helpful. I did understand that the reason they did that was to encourage us to figure it out ourselves, a mindset which would benefit us in the long-run; however, at that point when we were focused on solving the labs, these were not our thoughts. But one thing I enjoyed about these labs, is that despite all the challenges we faced, the satisfaction of having the robot do what we want it to do was immense. At the last lab, when I saw the robot working properly I felt proud that the efforts I put in through the semester.

The tutorials were also quite challenging as the theory lectures did not provide any practical or numerical examples on how to use them. In addition, the subject syllabus had been changed and my batch was the first to experience it. This rendered our only preparation resource, previous exam papers, useless. So we could only rely on tutorials which were challenging and insufficient.

The labs, though challenging, really helped me bond with my peers. We spent a lot of time doing the labs together and helping each other do the tutorials. The course was quite an enriching in that I learnt many things as well as bonded a lot with my peers and friends.

An important lesson this course helped me learn is the need for constant effort. Though the labs were difficult, we never stopped trying to solve them, partly because we had no choice, but also because we knew we would need the knowledge some day in the future. The reason we felt the satisfaction and pride at the end of the course was mostly because of all the effort than went into it. I think this is a very valuable lesson especially in college where everyone has a tendency to lose hope. This course helped me understand and feel the importance of continuous effort and learn that sincere effort will always yield good results.

Circuits