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.

URECA

Undergraduate Research Experience on Campus (URECA) is an invite-only research program for students at NTU who have shown academic excellence. The program helps students gain research experience at an early stage to encourage more students to take up research careers. The program lasts for one year with various deliverables along the way.

I was lucky enough, and of course worked hard enough, to be offered an invitation to participate in this program in my second year at NTU. I was initially confused as to whether I should take part in the program or not because of the increase in workload, given that I was already nervous about a few subjects in the third semester. I asked some people who had already signed up about why they wanted to do it. Most of the replies I got were related to gaining new experiences and building relations with professors. After collating some responses, I thought that building good relations with professors seemed to be a good enough reason to go for it. Additionally, gaining new experiences outside of coursework would also be helpful to build to my skillset and knowledge-base. So I decided to take up the opportunity and trust that I could manage the increased workload.

I met with a few professors whose projects interested me, and eventually chose the project titled <strong>“Deep Learning Based Network Classification with Little Training”</strong>. It aimed to develop a way to quickly and accurately classify large amounts of graph data without the need for a lot of hand labelled training data.

For the first meeting, there was just a basic introduction between my collaborator, my supervising professor and I. Our supervisor explained the basic objective of the project and a couple of terms; we also decided the schedule for our weekly meetings. The wheels were in motion and I was as excited as ever!

The first couple of meetings were more discussion based as our supervisor gave us various research papers to read and understand more about what our project was about. We asked many questions and received a lot of knowledge from our professor and research did not seem like such a far off fantasy anymore.

Once we were ready with the objectives of the project and had a basic grasp on the tools and models we were going to use, it was time to get coding! But even before that, we had to do tons (and by tons I mean tons) of data preparation. We had to take a dataset and randomly split it into 10 parts, and we had seven datasets! This task itself took us many days as we needed to write code to split the datasets and take enough time to make sure it was done accurately. The next step was to run each part through three different models and get the initial results. All this was the initial work, and we hadn’t even started working on the actual development we were supposed to do.

I began to realize the iterative and time-consuming nature of research at this stage. The conversion of the data from the form it was in the dataset to the form the models needed was the most tedious, time-consuming and boring job in the entire research project. The initial results also took very long to obtain because the models took very long to run as the datasets were huge. The initial results were basically labels given by SVM, RF and neural network models on testing data after training with different amounts of labelled data.

Our objective was to improve the quality of labels given by these models when trained with very little ground truth labelled data by using a lot of data labelled using our method.

After we were done with obtaining the initial results, we started coding on the actual method to label the unlabelled data accurately. This portion of the project was even more iterative than the previous one because we had to keep changing the code and looking for new and better ways to label the data based on Snorkel, a labelling model developed by Stanford University researchers. Snorkel was a fixed entity in our project as it had shown good performance in our attempts so far and it had easily accessible documentation and easy to use methods.

After a few months of hard work, we were able to come up with our respective methods which unfortunately could not meet our initial expectations, but gave us valuable insights regarding the problem statement.

Somewhere along the way, we had to submit a poster of our approach which is available on my <a href=””>LinkedIn page.</a> Poster making was an interesting experience as it gave us an avenue to be creative in explaining and displaying our methodology. The poster turned out quite well, and we were happy with it. Our supervisor also approved it in one go!

At the end of the academic year, we also submitted a full fledged academic research paper! Writing the paper was another interesting experience and required me to bring out all the English grammar I had learnt in school. We had to write individual separate papers even though we were collaborators and had to obtain approval separately from our supervisor. My paper turned out well as well and my supervisor was impressed by it! At the end we had a sweet little farewell email exchange and our supervisor praised our dedication and hard work.

All in all, URECA was a valuable learning experience and an excellent way to hone my collaboration skills. Also, at the end of the day, having research experience is also a shining star on my resume! The knowledge imparted to me by my supervisor, the teamwork with my collaborator as well as the individual work I did, will all be of great help to me in the future.

A must try for any NTU student!

Internship - MoveInSync

Amidst the disruption caused by the COVID-19 situation, I was stuck in Singapore for the whole summer of 2020. I had already completed 2 online courses, written a few more of these blogs and taken some much needed rest over 2 months. But I was in need of something to make the time I had left meaningful before the new academic year started. That is what led to me taking the decision of applying to an internship. I was able to secure a position with <strong>MoveInSync as a Computer Vision Intern</strong> after an interview.

MoveInSync is a corporate commute solutions company that provides a wide range of flexible services for corporate employees to manage their travel from home to work and back. The project I was assigned to was a new initiative being taken by the company to detect social distancing for workplace safety, given the prevalence of the COVID-19 virus. The internship was a short term engagement, for just over a month.

I was paired with a supervisor to work on the project. Contrary to my expectations, he was very relaxed and let me take my own path in searching for a solution to the problem statement; of course my expectations were not very accurate as this was my first corporate experience. Despite being very nervous and eager to exceed expectations, I started work calmly and took my time familiarizing myself with the domain I was working in and the type of work I was supposed to do. My supervisor was always very supportive of my achievements, big or small, and helped me debug a lot of my code when I was stuck.

Without revealing too many details, I would like to recount an incident during the internship where I realized tenacity and patience can solve any problem. As I mentioned the problem was detecting social distancing. Thus the bulk of the work was dedicated to detecting the distance between people. However, due to a number of factors, I was required to come up with a method to detect distance. I was excited to develop something new on my own. So I started searching for references on Google, but to no avail. The only things I could find were much simplified version of the problem statement and of not much help.

After about a week of nonstop effort, I was able to come up with a rudimentary approach to solving the problem. After a few tweaks here and there, it was ready in Python code and integrated with the rest of the program! However, there were many mysterious erroneous answers in the output. I was at a loss as to how that was possible as even my supervisor could not find anything wrong with the program. After another few days of searching, we found a small typo in the code and as it turns out that typo was the culprit of the wrong answers!

The reason I mentioned tenacity a while ago is that the net time it took to solve the detecting distance problem was half of the whole internship duration! Despite having a few limitations, I was eventually able to come up with a viable solution in the end. I was worried midway that I may not be able to complete it on time, and that the experience wouldn’t have been as meaningful. But because I was patient and did not get flustered or panicked, I was able to complete it well.

I also realized the need and importance of taking short breaks while working. I would not have been able to spot the typo if I had kept staring at the same code for hours on end. A pair of fresh eyes can do wonders while debugging code!

If you’re wondering what happened to the other half of the internship duration, then that was spent on solving some limitations of the solution and preparing to present the solution to the company CTO as well as other employees at MoveInSync. I prepared a set of slides as well as a live demonstration of the working of my solution. Including a short Q&A session, it was a proper corporate demonstration. They were all happy with my work and praised my solution.

They say “all’s well that ends well” and I think it is a very appropriate saying in my situation. Despite the first few days being stressful because of anxiety, I adjusted well and the internship was a memorable experience and helped me learn a lot of new things.

Despite the internship being 100% virtual since I could not travel to office due to COVID-19, I had a lot of fun working on the project and having daily calls with my supervisor, who gave me a lot of general tips on what I should do after my undergraduate studies at NTU.

I hope to get a chance to work with MoveInSync again!

Circuits and signals

The first part of this subject was about solving circuit diagrams using various rules such as voltage and current dividers, Kirchoff’s rules, and Thevenin’s and Norton’s theorems. I was quite comfortable with these topics except that the numericals were very long and tedious. Also given my reputation with careless math errors, I felt the need to be a little extra careful while solving them.

Most of the lab experiments for this course dealt with the first half as we were required to connect the circuits on physical breadboards and observe time constants and perform other calculations. This interested me quite a bit because there aren't many opportunities to physically observe things we only ever learn to calculate in books.

The second part was a little more theoretical in the sense that it dealt with signal processing and we had no physical lab experiments for it. It scratched the surface of Fourier and LaPlace transforms and how they are used to convert signals from the time domain to other domains for further analysis.

The final exam was also quite relevant to the syllabus and I was well prepared so I did quite well and achieved an A grade. The knowledge from this subject also proved to be quite helpful in a year 3 semester 1 course called Sensors, Control and Interfacing which built on the topics introduced in part two of the course.

Overall Circuits and Signal Analysis was an interesting subject with a lot of math and calculations but it was intuitive and relevant to other subjects as well.

Software Engineering

Another one of the major project based courses at NTU, Software Engineering focuses on a group project aimed at developing either an Android or a web application accompanied with proper documentation which is actually the main focus of the project. At the end of the development stage, each group is tasked to present their work to their lab supervisors.

The course did not have much teaching in the sense that the lectures were highly theoretical and only talked about the procedures to be followed while creating the deliverables (documents only) and did not teach anything about the actual development to be done. Thus it was a very “do-it-yourself” type course which made it quite stimulating in terms of time management and learning new things.

My group and I decided to play to all our strengths and develop a web application. We came up with our website idea and had it approved by out supervisor. Our final plan was to develop a currency rates portal where users can see the past and current exchange rates of selected currencies, watchlist them as well as locate money-changers using the Google Maps API. Our most important feature was the prediction of future currency exchange rates using machine learning based models. Additionally, we had administrative features to support any database changes that cannot be done by a normal user.

The deliverables included various documents such as a project plan, work breakdown structure document, use-case diagram and use-case description document, overall class diagram and sequence diagrams for the major features. We also had to create a video to highlight our work, and of course the source code itself.

The first roadblock hit us during the very initial stages of discussion. I was strongly in favour of using the SpringBoot framework in Java because it is an upcoming and popular framework used in the industry. Though I had never used it before, I was confident that we would be able to pick it up given that we were all comfortable with Java. However, another member wanted to develop it using the .Net framework because he was comfortable with it. This led to a slight conflict in the group as both of us were trying to push for our respective ideas.

Fearing this would lead to bad-blood in the group, we decided to present our ideas clearly to the other members and let them take the final decision. We proceeded with a whiteboard presentation in front of the whole group and finally my idea was selected.

The technologies we decided to use are:

1) SpringBoot framework backend with MVC design pattern

2) MySQL database

3) JDBCTemplate for database integration

4) Frontend in HTML and JSP

Once we had our tools in place, we divided the responsibilities and started work. Once we all knew what needed to be done everyone worked hard on their respective segments. I worked on the watchlist feature which involved all components of the MVC design pattern as well as using the database backend. Additionally, I created the video deliverable to explain the essential features and their working.

Apart from a few more small conflicts along the way, the project went smoothly. Though the final presentation had room for improvement in terms of explanation, I made the video clear enough to make up for it.

A part of everyone’s grades were also determined based on group peer evaluations. However, since my group was quite friendly and we resolved conflicts quickly, I was not too worried on that aspect. All of us managed to get a good final grade for the course and we were all very proud of our work.

I think Software Engineering taught me more about web development and group work than the documentation which was supposed to be the focus of the course. I learned how to peacefully resolve conflicts without taking them too far and how a degree of willingness to accept other opinions is very important to work well in a group.

Y2S2

I’m writing one article for this semester because not much happened after COVID19 set in so there isn’t much content to write one article per course.

This semester started off like any other with new subjects, classes and professors. The first two months were alright even though there was news about an infectious disease going around. However, by the time recess week came in March around it was already a requirement to wear a mask while travelling and constant sanitisation was a necessity. The rest of the semester after recess hardly had any studying as there was extreme confusion on the assessments and whether or not there would be final exams. Both the students and the university were both not ready for a pandemic so everyone was struggling to cope with the situation.

Eventually the university had to cancel all final exams and make the few quizzes we had given before that count towards our final grade with increased weightage. This decision came after a lot of changes in schedule which caused a lot of confusion among the students at NTU.

Given all this background, suffice it to say that the semester was very chaotic and stressful which impacted the mental state and ability to study of every student, including me. The university was gracious enough to take that into account and allowed us to change any and all courses into pass/fail grading, and were fairly lenient with grading as well.

In this light, the quality of learning in the semester was unfortunately heavily impacted. The subjects I took in this semester were:

1) Advanced Computer Architecture

It builds upon the concepts taught in Computer Organisation and Architecture from year 1 semester 2. It introduces more detailed concepts of assembly language programming through practical labs and informative lectures.

2) Digital Systems Design

It builds on Digital Logic taught in year 1 semester 1. The concepts covered deal with adders, multiplexers, LUTs (Loop-Up Tables) and other types of circuits. One especially interesting lab was where we were tasked to change the colour and resolution of a camera input using the above concepts.

1. Microcontroller Programming

One of the harder courses of the semester, this subject dealt with some basic Linux programming as well as programming in C on an Arduino Uno board to perform various functions such as multiprocessing.

1. Software Engineering

Given its group project-based curriculum, this subject was not affected much by the pandemic. Refer this article for details.

5) Marketing for the 21st Century

I took this subject under the General Education Requirements - Prescribed Elective in Business Management (GER-PE BM). It dealt with a lot of group work and in-class discussions. It helped me develop my soft skills and on-the-spot thinking. The structure of this class was quite different from other classes I had taken before. It was very vocal and group-discussion based and the professor encouraged us to discuss and find answers among ourselves. Every class, there would be a case study assigned to each group to work on for 15 minutes, followed by a discussion on the reflections by each group. This practice helped me improve my public speaking a little as well as helped with improving my collaboration skills. Additionally, we also had a final group project where we had to analyse any company, present a report on their business practices and come up with a solution to a problem we see with them. I learnt a lot about the business practices of Starbucks Coffee through this project! The experience of researching about something totally outside my usual area of study was also very interesting.

Though the semester ended in a strange way with the sudden cancellation of final exams, it was memorable for whatever I managed to learn through the stressful period. The semester was also followed by a 5-month “circuit breaker” period which was the lockdown in Singapore. The free time allowed me to reflect on my first two years at university, and plan for the next two.

Korean

My best friend introduced me to the world of Korean drama (K-drama) in 2015 which was much before it became popular around 2 years ago in 2018. I initially started watching my first K-Drama out of curiosity to see what she was so hyped about, and I ended up impressed with the plot (even though I find it lame today, but hey it was released in 2013 so considering that it wasn’t all that bad) and I was curious to check out some more shows.

Fast-forward to 2018 where I had watched about a hundred (not exaggerating) K-Dramas and avidly listened to K-pop (Korean music). I could also speak a certain level of Korean which I learnt by matching the frequently-used words from the dramas with the subtitles. At this point I was already deep into the world of K-drama and K-pop which made me want to learn the language even more to understand what my favourite stars are saying in videos that don’t have subtitles. I also wanted to learn the script because sometimes subtitles just don’t have the same effect as reading the native writing!

So I decided to teach myself the Korean script and broaden some of my vocabulary after my 12th grade board exams. Using some great online resources, it took me a few days to curate notes for the script and vocabulary examples but I was able to pick it up fast since I already had experience listening to it before.

By the time I entered NTU in August 2018 I was very familiar with the Korean script and could speak a fair amount with a good vocabulary. At this point, I wasn’t sure where I would ever use my Korean skills but it was something I was proud of because it added to the list of languages I could speak bringing it up to five!

When I saw my pre-registered course time table for year 1 sem 1, I decided I wanted to add a UE (unrestricted elective) for the semester. So while doing some research on that, I noticed that NTU offered Korean courses up to level 6. I immediately signed up for the level 1 class and excitedly looked forward to it.

On the first day of class, when the professor walked in and started speaking to us in Korean, I felt my excitement grow even more as I hoped to learn to speak like her, a native. Within the first few minutes of class I felt overqualified for level 1 as the professor started teaching us the alphabet which I had already learnt a few months ago. I then decided to have a chat with the professor at the end of class and tell her that I had learnt a few things myself and confirm if level 1 was the right class for me.

She invited me for a discussion to her office the next day where she asked me to briefly introduce myself and read a few paragraphs from a book, both in Korean. She was very impressed with my speaking and reading and suggested I move up to level 2. However, I was required to take an official placement test first, in which I did quite badly. I was confused to hear that I was too good to be in level 1 and not good enough to be in level 2; which is why I was told that I would need to put in a lot of extra effort to catch up with other students in level 2. I decided to take it up because I was genuinely interested so I was confident that I would be able to put in the effort needed.

I managed to do very well in level 2 which also had a presentation (more like a skit, which was a lot of fun to do!) as well as written and listening tests. I was expecting it to be much harder than it actually was, but that’s probably because I was very interested in it.

After the good results in level 2, I decided to continue taking Korean for all my remaining UEs. So in the next semester I took level 3 which taught me new kinds of grammar and gave me a chance to improve in my formal speech which was weak at the time. Additionally, it also had a lot of essay writing and an oral assessment which really put my speaking skills to the test. I also met one of my closest friends at NTU in level 3 who also directly entered in level 2, with whom I often speak in Korean.

The last Korean level I was able to take at NTU was Korean level 4 in year 2 sem 2. I was out of UE credits by then so I had no choice but to end there. Level 4 was of course the hardest level which introduced many more aspects in writing and more nuanced speaking. By the end of level 4, I was also able to understand other people speaking Korean faster and more accurately with the help of the listening exercises over all three levels I took.

I’m proud to say that in all three levels of Korean, the professors and my peers praised my speaking and pronunciation by saying I sounded almost native. Thanks to the Korean I learnt at NTU, I learnt more about the Korean culture and etiquette. I can also watch most K-dramas without subtitles (except those with jargon) and understand the lyrics of many K-pop songs!

Although my journey of learning Korean started for a different reason, by the time I finished level 4, I became more interested in the Korean culture and traditions than I was before and it wasn’t just about K-drama anymore.

Another important thing this whole process taught me is that I actually have a flair for learning languages! This is a talent I genuinely want to work on going forward so who knows which language I will pick up next?

MDP

Multidisciplinary Design Project, or MDP for short, is the only course in the whole Computer Engineering (and Computer Science) degree which is worth four credits; all other courses are only worth three or fewer credits. This is proof that it is one of the hardest courses in the degree and it definitely lives up to its name.

Before talking about my experience, let me first elaborate on the premise and objectives of MDP. I’m doing this first because it will help to understand the incidents I recount in the following text.

Essentially, MDP is a project-based module done in teams of 10 members consisting of both Computer Engineering (CE) and Computer Science (CS) students. The whole course kind of follows a “do-it-yourself” mindset where there are no official lectures and the only support students get are some briefing and technical guides.

The final objectives of MDP are to build a robot that can:

1) Fully explore a maze of given fixed dimensions

2) Find the fastest path from the start zone to the goal zone of the maze while passing through a waypoint (a fixed point on the maze, set on the spot during the competition)

3) Correctly recognise images placed in the maze along with their coordinates

“The maze” is basically the area the robot needs to explore while avoiding obstacles and recognising images.

The above are the three competitive challenges according to which all MDP groups are ranked. A group’s ranking determines a certain percentage of each group member’s grade. Each challenge is a leaderboard of itself in which the groups are placed in order of performance in the challenges.

The components needed to be developed to achieve these tasks are:

1) Actual robot assembly and coding functions for motors and sensors (Physical components connected to an Arduino board which is in turn connected to a RaspberryPi via serial port)

2) Algorithm development for exploration and fastest path (run on a laptop connected to the RaspberryPi via WiFi)

3) Andriod module to control the robot since we cannot use laptops while the robot is running (Android tablet connected to the RaspberryPi via bluetooth)

4) Communication channel to coordinate the three components and pass messages (the RaspberryPi itself)

5) Image recognition module (can be either in the RaspberryPi or a second laptop connected to the RaspberryPi via WiFi, we did the latter)

Additional deliverables include a project plan document and clearing a functionality checklist before the competitive challenges. Each group also has to submit a video and write a wiki blog highlighting their work at the end of the semester.

Now, on to my experience of MDP!

I had first heard about MDP in year 2 sem 1 as this huge threatening mountain of work all third/fourth years have to do. I didn’t know much about it except that I had seen seniors working on a robot all day in labs before but I had no idea what they were doing. So I was fairly clueless and nervous but also a little optimistic about what it would be like.

The first day of MDP was on a Friday morning at 8.30 AM in the lab. Before that, each student was assigned to a group at random and each group was assigned a lab to come work in. The first shock I received when I checked my grouping assignment was that I was the only CE student and the rest were all from CS; which basically meant I was the only one who had any experience in hardware and I would have to directly handle the robot. This was a scare for me because the only extra experience being a CE student gave me was some coding experience on the Arduino board and nothing on how to actually assemble a robot. But I decided to cross that bridge when I got there.

The first lab was more of an introductory session for us to get to know our teammates with who we would be working with for the rest of the semester. We received our kit, nominated a group leader and MDP AY2020/2021 kicked-off!

We started with a meeting to divide the tasks. I thought being open about my strengths and weaknesses would be better than biting off more than I could chew at the start and underdelivering at the end. So I told everyone that as a CE student, I do not get experience in robot assembly and that I am as new to this as the other CS students in the group. They appreciated my honesty and assigned extra manpower help with the robot.

I worked as part of the Arduino team with two others who were brilliant students and did a lot of excellent work despite not having any experience in hardware. We started assembling the robot with the help of a very short guide (that only tells how to fit the wheels to the main body) and very useful user manuals of the components which had schematic diagrams. Within a few hours we were able to assemble the wheels and the Arduino board. [pic]

I knew how to write basic programs using the serial port on the Arduino board but I had never programmed wheels using the given peripheral boards (microprocessor programming dealt with programming the wheels of a robot, but it used a different approach altogether and so was not of any help here). So we had to write our own logic to make the robot move as we needed it to.

The main trouble of working with the motors was designing a PID controller to maintain straight line and accurate motion. Various concepts were involved in designing the controller and it involved a lot of calculations and trial-and-error for parameter tuning. It was during this process that the Arduino team started spending whole days together in the lab trying to figure out how to make the robot go straight.

It took us about two to three weeks to complete the motor functionality including straight line motion and accurate rotation (for any degree such as 90 or even 1080). we also finished working on the sensors. However, we noticed a lot of unreliability in the hardware that kept changing the calibration of both motors and all six sensors. We had to spend a lot of time trying to recalibrate and fix our motion and sensor readings.

At the same time, the other teams were working on their tasks as well. However, there was a delay on the algorithm side as they were working on some documentation deliverable towards the beginning. Thus as the deadline for the first challenge drew near, there was a lot of pressure on them to finish creating the algorithm for us to integrate with them and test it out. The other teams had completed most of their tasks as well.

By that time, the robot was completely assembled and ready to go. [pic]

By the time recess week came, every group was in a state of chaos under the pressure of the upcoming challenge in one week’s time. Additionally, the COVID-19 restrictions were also preventing many group members from coming to the lab to help out. Our group was also worried by the continuous needs for recalibration delaying our checklist clearance and the delays caused for algorithm development.

After clearing the checklist, two of the group members from other teams stepped up to help out with the algorithm and eventually managed to complete it on time. However, we lacked enough time to test before the first run. However, we tested as much as we could and adopted a “add-as-you-go strategy” for extra functionality such as corner-calibration, emergency braking and alignment of the robot needed for the challenges.

Despite being short of time, we managed to top the leaderboards for both exploration and image recognition for the first week. It was a completely new experience watching out robot explore the maze as we cannot control it in anyway and can only hope it completes successfully without hitting an obstacle. Despite doing well in the other challenges, we failed to clear the leaderboard for fastest path.

There was much more work left to do for the next iteration of the challenges as the battery we were using for our robot seemed to have become damaged with an increasing voltage upper limit and faster draining time. We managed to make it to the second run but again failed to clear the fastest path leaderboard due to a calibration miss.

After changing the battery for the last run, we again could not clear the fastest path leaderboard because we failed exploration that week due to the persistent battery problem.

Unfortunately as the maze becomes harder every week, we were eventually not able to clear the fastest path leaderboard. However, we came third in both exploration and image recognition which we were all proud of achieving.

I’d also like to mention that the university was kind enough to give us extra lab contact hours after some MDP student representatives put in a request for it. The evening hours we spent in the lab working on the robot made me feel like a real university student working hard and I realised that I had become the senior working on the robot all day everyday. It was a moment that made me realise I had come very far in NTU from my days in first and second year observing my seniors working on that mysterious robot. One of my juniors even pointed out to me that he sees me in the lab working on this all the time :-)

Once the leaderboards were done, the pressure lifted by quite a lot as we only had the wiki blogs and video left to submit. Both of these went smoothly as well and led to the conclusion of MDP with the disassembly of the robot. We even took a commemorative shot of the components after disassembling it![pic]

Most of my year 3 sem 1 went by with me working on MDP in the lab. I spent almost three days every week from 8.30 in the morning until the lab closed at 5.30 PM working on the robot in the lab. Many breakfasts and lunches were sacrificed, but many friends were made instead. It was extremely pressurising and stressful but, it was rewarding to achieve whatever we did and I was proud of the team and myself for finishing strong without any major conflicts.

Additionally, the other groups assigned to my lab were also very helpful and cooperative. All groups helped and received help from each other; we also created a queue system to manage the usage of the maze for testing!

MDP is something every CE and CS student at NTU has to go through and though it seems daunting at the start, working on one task at a time and trusting my teammates is what got me through it. I also realised that is is important to accept that it is not possible to oversee every aspect and communication and feedback within the group is essential for success in such a huge project. The continuous effort we put in and the perseverance we show are the only factors that can affect results we achieve.

MDP made my semester hard but also fruitful and it is an experience I’ll never forget even after I graduate from NTU.

Machine learning blog

Machine Learning was one of the technical electives I had taken in the final semester at NTU. It dealt with basic machine learning techniques and taught concepts such as decision trees, generalisation and Bayesian networks in the first half and concepts such as SVM, random forests and dimensionality reduction in the second half.

The first half of the module was primarily based on more traditional machine learning methods that did not involve neural networks. Concepts such as decision trees were inference based and the objective of teaching us these concepts was to intoduce us to the evolution of different techniques and their pros and cons instead of directly diving into the neural networks and other models. The concepts in the first half were relatively easy to understand as they were intuitive. However, I faced some difficulty with the Bayesian Belief networks since they were not as intuitive and required a lot of practice. Eventually I got better at it as the exams drew near by consistently practicing the past year paper questions repeatedly to get a hang of the approaches.

The second half of the module dealt with more advanced methods such as SVM and random forests. Though those concepts were simple enough to grasp, the later topics such as dimensionality reduction and PCA were relatively much harder, needing many late night study sessions with my friends who were equally struggling.

There was also a course project which constituted 40% of the overall grade with the remaining 60% allocated to the final exam. We were supposed to form a team of 4 members, select a project from the list of options provided and work on it starting week 4 of the semester. I made a group with three people I knew out of which one was a friend I was quite well acquainted with from the first year as we were neighbors for the whole 4 years of university. The topic we selected was the “New York Taxi Fare Prediction” Kaggle challenge as it seemed to be a relatively simple regression problem statement which we were all confident in. We divided the tasks in pairs and I worked with my friend for the tasks.

Looks can be deceiving, and the project was more complex than we first thought. There were not many features provided to us and a large part of our efforts went into feature engineering to improve the model performances. After performing a few basic tests on simple models such as SVM and random forests from ScikitLearn we decided we needed at least 4-5 more features to get sufficiently good performance. We referred previous students solutions, YouTube videos and other methodologies to get ideas on which features we can add as well as which model would be best suited for the purpose of the project. The features we added include Haversine Distance, Bearing Distance, Airport Distances and others. Also, we selected the XGBoost and LGBM models as they showed the best performance over the other options such as neural networks and SVMs.

The dataset we were to use had 55 million rows! This meant a lot of computing power or time would be needed to get the final results. After several attempts on Google Colab we found that loading all the data at once would not be possible. Thus we researched on several methods such as loading the data in chunks to load our data into the runtime. However, none of the methods was successful. We also lacked the time to experiment on further data loading methods since we were balancing this assignment with other modules, projects and our FYPs. Also the project deadline was the day before the final exam which we all needed to study for. Thus we decided to optimise memory usage by combining the models in series. We loaded in a random sample of 22.5 million rows to train one model, deleted the data and loaded the rest of the rows to train another model and then combined the results of the two models using a weighted average ot get the final results.

We placed high on the Kaggle leaderboard and performed well in the overall statistics of the batch.

The final exam was about 17-20 hours after we finished submitting the project. It was extremely hectic to quickly wrap up the project, report and record the video presentation while revising for the exam. Although the exam was open-book, we all knew that meant the questions would never be directly from the lecture material. Watching videos on YouTube to revise the concepts which executing the codes for the project was a normal for us at the time. After submitting the project, we quickly started going through each chapter, solving the past papers and discussing the answers.

Eventually, the exam went well and I achieved an A grade for the module.

Information retrieval blog

Information Retrieval was another technical elective I had taken in the final semester. It was divided into three assessment components- the first being a quiz, the second was a project and the third one was the final exam. The quizzes were of low weightage but the project was 35% making it quite a high-weightage important project.

The concepts dealt with were different methods and algorithms used for information retrieval such as the boolean retrieval, biwords, levenshtein distance, tf-idf algorithm, web search, andv arious methods of evaluating search accuracy such as precision, mean average precision (MAP) and others.

The course content was quite heavy and theoretical and needed a lot of memorisation especially because the final exam would ask theory questions and we were expected to answer them directly from the lecture material. The lectures themselves were very intensive as many concepts were fit into a single two hour lecture. Understanding the topics was not as hard since the basic idea behind them was quite simple.

The project was a group based and needed us to choose our own groups of 3-6 participants to work on the project. I had taken the course along with a friend but since we did not know other people, the rest of the group was randomly allocated. Our group finally had a total of 5 members with 4 final year students and one third year student.

The objective of the project was to develop an information search system based of a self-defined problem statement. We had to use Apache Solr as the information storage and retrieval system and build a web interface to search for and display the information. Additionally, we also needed to perform sentiment analysis using machine learning models on the data that was retrieved. The data itself was to be obtained using a web crawler.

The first group meeting went well with all the group members proactively voicing ideas on the tech-stack and kind of data we can base our project on. I was beginning to consider myself lucky to have another good random group allocation. However, by the second meeting it became clear that my friend and I would need to take the lead on the project since the group members seemed very reluctant to work on the information retrieval system or the web application. Thus we divided the tasks such that we would work on those two components and the rest of the team would work on the data crawling and sentiment analysis portions.

As we were also new to Solr, we decided to do some research on it before starting our code. Referring to the Solr documentation helped us gain some confidence with using the system itself and previous years projects gave us an idea of the kind of system we needed to build. For the web application we decided to build a simple Flask application with ReactJS as the frontend. My friend took charge with the frontend and I took the responsibility of the backend and configuring and setting up Solr. We worked for days together to make the system work while also helping the rest of the group with their concerns. We also simultaneously worked on the report for submission.

After three weeks of hard work, we had built our information retrieval system and completed the report as well. After many rounds of final checking we submitted the project 3 minutes before the deadline!

Since the assignment deadline was about two weeks before the exam, we had a few extra days to study for it, unlike the situation with the Machine Learning assignment. This was only one of the first deadlines of the semester. I had to manage the machine learning assignment, studying for three exams and the FYP load despite having submitted this project.

In spite of the situation, I managed to do well in the exam, which stayed true to the pattern indicated in the past papers which gave me a good idea on what to expect. The hard work paid off as a final A grade.

Dsp blog

Digi comms blog

Computer vision blog

Nlp blog

Advanced algo blog

Sensors blog

Sensors control and interfacing was an extension of circuits and signal analysis from second year. It was an interesting subject in that it was very relevant to things we use in real life. Part one of the course dealt with sensors, for example RTDs (resistance temperature detector) and operations on the signal in analog domain. The content taught included numerical examples on solving the different parameters of the sensors and on other operations and properties such as amplification and signal-to-noise ratios.

Due to the COVID-19 situation, there was no final exam also the content was tested in two quizzes- one for part 1 and another for part 2. The format of the quizzes was made to be similar to the final exam and so practicing from past papers was a good way to study for them. The questions on the first quiz were very similar to the past final exam questions so I managed to do very well in it.

Part two of this course dealt with operations on signals in the digital domain. Thus the content covered Laplace and Z transforms, system stability conditions, and controller design. All the topics were interrelated and quite mathematical, and involved a lot of algebra to solve for a system transfer function, check for its stability and find the best controller parameters.

The quiz for the second part was very different from the final exam format and was a surprise for me since I had prepared only by solving past papers.

Additionally, 50% of the final grade was determined by lab work. It was to be done in pairs and the task was to design an amplification circuit for the output of a pulse plethysmograph. A good portion of the grade was also determined by the implementation of an additional feature which each group had to come up with on their own.

I was paired with a friend who I made worked with before and we both had a similar approach to the tasks so we worked well together. We were given an arduino board, resistors, capacitors and ICs to work with for the project. The final results had to be displayed using matlab.

We were both weak in circuits so we faced some difficulty in the project. After a lot of perseverance, we managed to achieve the correct results. After completing the basic task, we had to move on to the additional feature. We chose to make the additional feature in matlab instead of modifying the given circuit.

Our additional feature was twofold-

1) calculate heart rate

2) Play audio alerts in 4 different languages for heart rate high/low/normal

It was a unique feature that no other group had done and if was fun to make as well! We recorded the audio alert messages by playing the messages on google translate. Then we integrated it with the rest of our matlab program.

We also had to present our understanding of the circuit for an oral assessment. We prepared slides and rehearsed for the presentation which was in a casual setting with just the professor and us. The professor asked us questions on the different parts of the circuit for example what would happen on switching or changing certain components. We were able to answer the questions correctly and did well for the assessment.

Computer networks blog

Neural nets blog

Final ntu blog