

2018

NUS School of Computing

STINT Teaching Sabbatical @ NUS School of Computing

A REPORT ABOUT MY EXPERIENCES AT NUS
WILLIAM JOBE, UNIVERSITY WEST

Introduction

This report is a summary of my STINT teaching sabbatical at the School of Computing (SoC) at the National University of Singapore (NUS). This summary focuses on my teaching duties and other pedagogical activities at NUS, however, it even addresses some practical and personal experiences of participating in a STINT teaching sabbatical at NUS and living in Singapore with a family. During both the Spring visit and the actual teaching sabbatical I was accompanied by my wife, Anna Jobe and two of my children: Emma Jobe, 15 years old and Isaac Jobe, 12 years old. The structure of the report is chronological with separate sections dealing with my pedagogical and personal observations of working at NUS and living in Singapore.

Spring Visit/Planning – March 2018

The STINT teaching sabbatical adventure technically starts with the notification of receiving the STINT scholarship before the Christmas break as well as the kick-off conference in Stockholm in February. However, the true start occurs once you have booked and made the Spring visit to the hosting institution, NUS in my case. Soon after the notification of being awarded the scholarship, you are provided with a contact person. In my case I was first provided with two contact persons at NUS, Dr. Min-Yen Kan and Mrs. Susanna Lam. Dr. Kan worked at the School of Computing (SoC) and was my key contact person regarding my teaching duties. Mrs. Lam worked at the central administration at NUS and was responsible for all other questions regarding my stay. Mrs. Lam, in turn, delegated all administrative questions to Mrs. Wendy Wong, who worked at the administrative department at SoC. All of the planning for the Spring visit as well as my stay during the fall was, thus, handled by Dr. Kan and Mrs. Wong.

Through a number of email conversations I was able to quickly plan my visit to SoC and NUS. Dr. Kan helped to arrange meetings with key academic personnel to plan my teaching duties and Mrs. Wong helped book an apartment to stay in as well as provide me with contact information for local international schools so that I could arrange my children's schooling. The dates of my Spring visit were from the 25th of March until the 1st of April 2018.

My thoughts regarding my Spring visit is that the trip is key for outlining your teaching duties as well as preparing yourself and your family for life in Singapore and at NUS. Once we arrived, I was surprised that NUS had arranged an office with a sign for my short visit. During my stay I was able to plan for primarily participating in two computer programming courses (though changes were made later) and make arrangements for my children's schooling. My kids had to, for example, take entrance exams before being able to officially apply to international schools in Singapore. Additionally, we were able to test our accommodations, the local cuisine/stores, and even do some sightseeing in order to get a feel for living and working in Singapore. After the visit we returned to Sweden with the feeling that our stay in Singapore was going to be a great experience both for me and my family.



FIGURE 1 - VISITOR SIGN ON MY OFFICE DOOR

Fall Stay and Working at NUS

On the 1st of August 2019 my family and I flew to Singapore and moved in to our home away from home at Kent Vale housing, a three-bedroom apartment on the 12th floor of a 25 floor, high-rise building. The apartments were fully furnished, and it was roughly a 15-minute walk to my office. My actual teaching duties and the children's school started on the 7th of August, so we had a few days to get settled in and get over our jetlag.



FIGURE 2 - LIVING ROOM IN OUR APARTMENT



FIGURE 3 - VIEW OF OUR APARTMENT BUILDING FROM THE POOL

The first two administrative hurdles to overcome were getting my work pass and my family's dependent passes from the Ministry of Manpower.



FIGURE 4 - THE WORK AND DEPENDENT PASSES

Mrs. Wong at SoC was instrumental in streamlining the process and paperwork so all we had to do was complete the forms, show up at the ministry to take pictures and fingerprints, pay the fees and finally collect our passes. The work and dependent passes are important as you need them to reenter Singapore if you travel internationally during your stay as well as you can receive a variety of

rebates using the card. Additionally, Mrs. Wong even helped arrange my staff card so that I could enter all the different lecture rooms and computer labs.

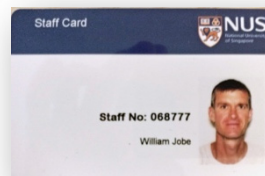


FIGURE 5 - NUS STAFF CARD

Before I dive into the gritty details of my teaching duties and experiences at NUS, I think that is important to have a general understanding of NUS, SoC and studying at university in Singapore in general. NUS is a world leading university and internationally ranked top 25 on most lists and ranked number two in Asia. NUS consists of 17 different schools which focus on a specific subject area and range from subjects such as Law, Medicine, Engineering, Business and of course, Computing in my case. Each school in turn consists of departments and the School of Computing has two departments: the Department of Computer Science and the Department of Information Systems and Analytics. I worked at the School of Computing and was involved in two modules, one from the Computer Science department and one from the Information Systems department.

NUS has a Fall and Spring semester. The Fall semester is from August until December and the Spring semester is from January to May. A full-time student typically studies 20 credits or five modules (the NUS name for a course) worth four credits per semester. All five modules are studied in parallel and most courses have at a minimum a mid-term and final exam. Each semester is 18 weeks long and includes two recess weeks before the midterm and final exam weeks. The standard NUS academic calendar is shown in the table below.

Orientation Week	1 week	---	Commences on first Monday of August each year. The academic year starts off with the Freshmen Inauguration Ceremony.
Instructional Period - part 1	6 weeks	6 weeks	In Semester 1, this starts in the week immediately following Orientation Week. For Semester 2, this commences on the second Monday of January each year.
Recess Week	1 week	1 week	From Saturday to the following Sunday.
Instructional Period - part 2	7 weeks	7 weeks	Classes resume on Monday immediately following the Recess Week.
Reading Week	1 week	1 week	Lasts from Saturday of the final week of the preceding Instructional Period to Friday the following week.
Examination	2 weeks	2 weeks	Starts on the Saturday immediately following Reading Week.

FIGURE 6 - THE BASIC STRUCTURE OF A SEMESTER AT NUS

My Duties and Responsibilities

I chose during my Spring visit to focus on being involved in the Freshman (first year) programming courses and choose originally two courses from the Department of Computer Science, one in Python and one in JavaScript. I quickly learned that these two courses were more or less copies of one another, and the only real differences were the students taking the course and the programming language. I researched the different freshman programming courses taught at NUS. There are six fundamental programming courses: CS1010, CS1010E, CS1010J, CS1010S, CS1010X and CS1101S. CS1010 is the standard course and it along with CS1010E (for Engineers) and CS1010J (for Information Systems majors) are taught in C and Java respectively. These standard courses do not seem to be gamified and follow a slower pace. The three remaining courses CS1010S, CS1101S and CS1010X all used game-based learning in some way. The two S courses are fast-paced and use the same curriculum (and are the original two courses I intended to take part in). The CS1101S has been the elite programming course (students were accepted only on grades and an interview) for a number of years, but it is now obligatory for all of the Computer Science majors. The CS1010X is an online only version that is targeted towards prospective students (or the general public) before they start at NUS. I thus switched the course in Python (CS1010S) to the Java course (CS1010J) that was given by the Department of Information Systems instead. This allowed me to take part in an “elite” programming course that used a declarative (functional) programming paradigm as well as their Java course that used an imperative (procedural/object-oriented) paradigm in order to be able to compare and contrast pedagogies and learning strategies in distinctly different programming courses. Therefore, my primary duties and responsibilities were in the CS1101S and CS1010J modules.

General course structure and pedagogy at the School of Computing

Before I describe in detail the work, I performed in my two courses, I will describe the general structure and pedagogy used at the School of Computing, specifically in their programming methodology modules. The primary forms of face-to-face instruction in the modules are lectures, reflections and studios. Lectures are the standard auditorium lectures where the teacher is on stage and all of the students attend and listen. The lectures are typically two hours in duration and have limited interaction with the students. Reflections are typically one hour in duration and done in smaller groups, roughly 20-30 students depending on the module. Reflections are an interesting mixture of lecture and lab. The reflections are taught by academic staff and focus on “lecturing” on specific concepts and problems from the actual lecture. Reflections are much more interactive than lectures and ideally the students talk and participate as much as the staff leading the reflections. Studios are typically two hours in duration in groups of eight students and taught by teaching assistants (usually older students). Studios are supervised problem-solving sessions, where the students are given problems in advance to work on and the teaching assistants act as mentors/guides to assist the students if they cannot solve a particular problem. In summary, a typical programming methodology course has five contact hours/week.

Programming methodology modules at the School of Computing use a variety of assessments. Reading assessments use multiple choice questions (MCQs) to test factual knowledge about various concepts. The midterms and finals typically require students to write programming code by hand and practical exams require students to solve programming problems using a computer in a controlled lab environment. The final grades in a module are determined by weighting the performance on the various assessments and homework as well as attendance at reflections and studios. The weightages for the final grade vary from module to module and grades are given using the A-F scale. Some modules even use the Bell Curve to determine the final module grades.

My courses

The following section generally describes and highlights my duties and responsibilities in the CS1101S and CS1010J modules as well as my shadowing of an IT governance course in the Information Systems department. If you want to read all the specific details of my time at NUS, then you can follow the blog that I maintained during my teaching sabbatical. The blog was updated many times each week for the duration of my stay and in total I made around 120 blog entries. This report is, consequently, more a summary of my activities and reflections of my experiences. My blog at NUS can be found here:

<http://blog.ei.hv.se/williamjobe/blog/>

CS1101S

As previously stated, the CS1101S module was the “elite” programming methodology course and this course was the main focus of my time at NUS. Specifically, I attended all the lectures/briefs, taught three reflection sessions each week, participated in the weekly staff council meetings, gave personal consultations, helped grade some missions (homework), helped set the first reading assessment, and helped grade the midterm and final exams. The following section describes in detail my aforementioned duties as well as all the other noteworthy aspects of the CS1101S module.

Instruction and pedagogy

The pedagogical philosophy of the CS1101S course was very clear from the start and took inspiration in social constructivism in that learning is done together and is actively constructed by an individual learner. The article *The Case Against Teaching* by Larry Spence (2001) was shared before the start of the module as inspiration to design learning experiences instead of teaching and this was a common theme throughout the implementation and execution of the various weekly instructional interventions described in the following sections.

Each week there was a two-hour lecture on Wednesdays and a one-hour brief on Fridays. The lectures progressively introduced a new topic and the briefs typically presented topics tangentially related to the course topics to arouse general interest in programming and computer science. Some of the topics were, for example, graphics and sound processing as well as 3D video. The



FIGURE 7 - AN EXAMPLE OF A LECTURE IN CS1101S

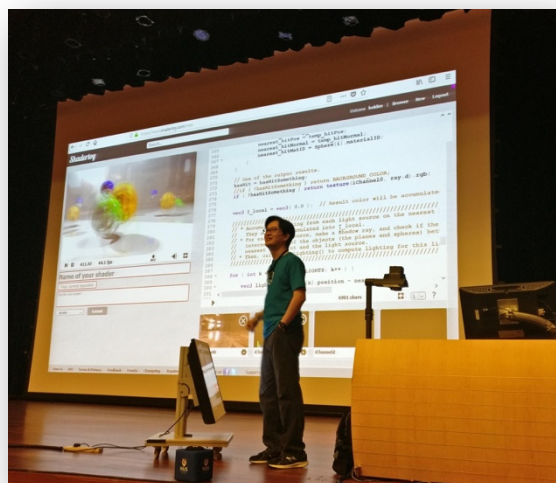


FIGURE 8 - AN EXAMPLE OF A BRIEF IN CS1101S

lectures and briefs were taught exclusively by Dr. Martin Henz and Dr. Low Kok Lim. Attendance (there were around 450 students enrolled in the module and around 425 by the end of the semester) at all the lectures and briefs was very high given that the lectures and briefs were recorded and made available on the course's LMS afterwards. During each lecture a path was made available on the course programming environment which was called the Source Academy (see the ICT tools section below for a detailed explanation). The path was a simple multiple-choice quiz that the students answered during or after the lecture. The paths were not graded and were an attempt to increase the interaction and engagement of the lectures.

The reflections took place each week between the lectures and briefs, so it was Wednesday afternoons and all day on Thursdays. Thursdays were thus my busiest days and that was when I taught my three reflection sessions. From my Swedish perspective the reflection sessions were a type of lecture/lab hybrid. I both lectured on specific topics but also tried to engage the students as much as possible by allowing them to show how they had solved the various programming problems. Because the reflections were in smaller groups (around 20 students in my reflections) I was able to get to know the students and develop a relationship. This obviously helped pedagogically as was able to see which students were struggling and which were doing well in the module.

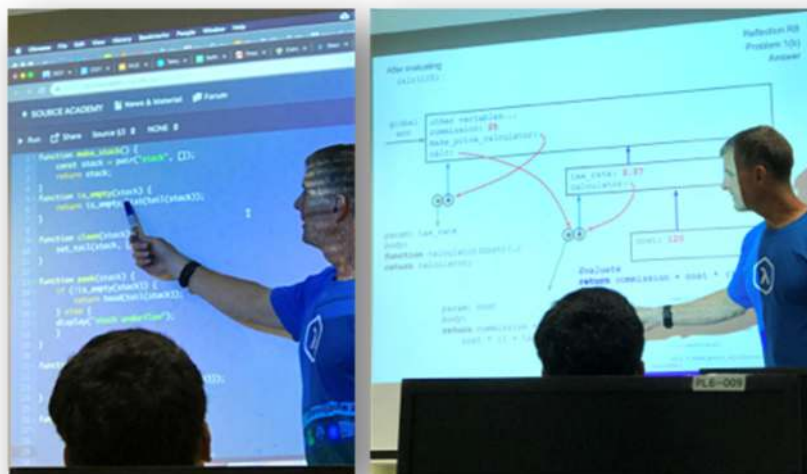


FIGURE 9 - ME TEACHING SOME REFLECTIONS

My participation in studios was limited to a form of quality control. Due to the fact that the teaching assistants (called Avengers in the CS1101S module) are previous students and that the CS1101S had scaled up to be an obligatory module for all CS students, there were a number of CS1101S avengers who had not taken the CS1101S course themselves. The senior academic staff was thus worried that the quality of mentorship/guidance in the studios might be lacking. Therefore, in the beginning of the semester I, along with the rest of the academic staff, was assigned a number of studios to monitor the quality and provide feedback to the teaching assistants afterwards. On the whole, the teaching assistants in the studios that I monitored were ambitious, engaged and well-prepared. They truly wanted to help the students succeed. However, some issues in studio quality were detected, but these issues were quickly addressed by the senior academic staff.



FIGURE 10 - AN EXAMPLE OF A STUDIO SESSION

The final form of instruction that I was involved in was personal consultations. Personal consultations are something that any student can request from a member of the academic staff. Typically, a student will contact the staff member who teaches a student's reflection session. A personal consultation can be viewed as a longer and more structured reflection session that is given in a one-to-one or one-to-two fashion. Obviously, after each reflection session, students ask questions as is always the case. However, personal consultations provided a formal outlet for those students who needed more help. I gave a number of personal consultations, primarily with two students, and most of the consultations took place in the second half of the semester.

The final activity and duty that took a great deal of my time and effort was assessments. As previously stated, I helped set the first reading assessment. The means of assessment at SoC were very different and more numerous than my previous experience at University West. In short, I learned that I have little, practical

experience making multiple-choice questions and exams. At University West we have small class sizes and MCQs are not frequently used, but at SoC, MCQs are a necessity due to the scale of the obligatory modules where hundreds of students per module is common. Nonetheless, I learned a great deal in the process of making an MCQ and developed a better understanding of the value of MCQs. I also assisted in grading a mission (homework) in the beginning of the semester because the auto-grader function was not working as intended.

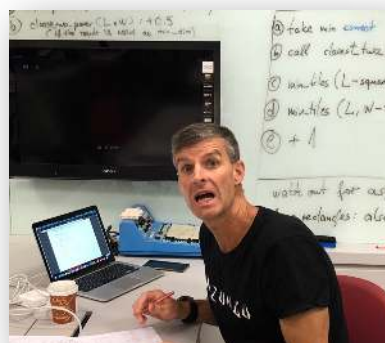


FIGURE 11 - THE FIRST READING ASSESSMENT WITH MCQs IN CS1101S

The most daunting tasks in the module were grading the midterm and final exams. Both exams consisted of a number of programming questions in which the students wrote the answer by hand. In other words, it was a time-consuming, manual process to grade both the mid-term and final. Additionally, the midterm and finals were only graded by the academic staff, so manpower was more limited. It took the academic staff roughly six days to grade the exams working from 8am to 10pm every day, weekdays and weekends. Once again this was a personally challenging process as we very rarely have students in programming courses as University West write code by hand anymore, so I was very out of practice. Students at University West typically are graded by practical examinations where they write solutions at a computer in more authentic settings. However, I can now appreciate the mental rigor required to solve solutions (and grade them) without the help of an interpreter as it is an acquired skill like any other.



FIGURE 12 - MIDTERM AND FINAL GRADING



Administration

As with all university courses there are always administrative tasks that must be handled and the CS1101S course was no exception. Because of the size and ambition of the course, in addition to the typical administrative tasks associated with a course, the CS1101S module had weekly staff council meetings on Tuesday evenings throughout the semester to deal with all the various day-to-day issues with quests, missions, assessments, lectures, reflections, studios, and personnel as well as plan and delegate the preparation and execution of all the different activities for the coming weeks. The staff council consisted of the academic staff as well as 10 senior avengers, and I participated in nearly all the staff meetings which provided great insight into the inner workings of the course and teaching at SoC/NUS in general. An interesting item that arose during the staff council meetings was the “medical center and triage”. The

academic staff was very proactive in finding and designing specific pedagogical interventions for students who were struggling, which was called the “medical center”. The progress of students in the medical center was tracked and “triage” was performed to help the students with efforts such as extra lectures/reflections and personal consultations.

ICT Tools

Nowadays all modern teaching at the university level involves the use of a variety of ICT tools and NUS/SoC are no exceptions. In order to perform at a reasonable level in the courses I worked with you must have a solid grasp of ICT tools and general and be able to quickly assimilate and use new tools. A deep understanding of ICT was a prerequisite and fortunately for me it was never an issue, but I can envision a scenario where it could be. Furthermore, the students were required to possess the same ease of using and learning new ICT tools and it never seemed to be an issue. Compared to University West the diversity and number of ICT tools used in the CS1101S course was surprising and I am skeptical that students and/or staff at my home university would have been able to so easily adapt to the different ICT programs and systems used.

The first and most ubiquitous software was the LMS, of course. At NUS there were currently two LMS systems which were developed in-house. The older system was entitled Integrated Virtual Learning Environment (IVLE) and the newer system was entitled LumiNUS. NUS was currently in the process of migrating from IVLE to LumiNUS. Originally the CS1101S course intended to use the new LMS, but it was slow and buggy and did not have all the functionality desired for the course, so the older IVLE LMS was used instead. IVLE offered standard LMS functionality and was used for sharing resources with students, tracking groups and attendance, mass communication, etc. For questions and answers the Piazza forum tool was used. Piazza was a powerful forum application that I plan to use at University West in the future. For the staff and administrative tasks, the Telegram messaging program was used for communication and Google Docs was used for the sharing of resources.

However, the most prominent and important ICT tool used in the CS1101S course was the Source Academy (<https://sourceacademy.nus.edu.sg/>).

The Source Academy was both the actual programming language used in the module called the Source (a limited version of JavaScript), an online integrated development environment (IDE) and a gamification platform. In the Source Academy students followed a game-based scenario through the course material, did their homework missions, paths and quests, and basically wrote all their programming code. The Source Academy is a work in progress that has been iteratively developed by previous students and staff over the last few years. The actual implementation

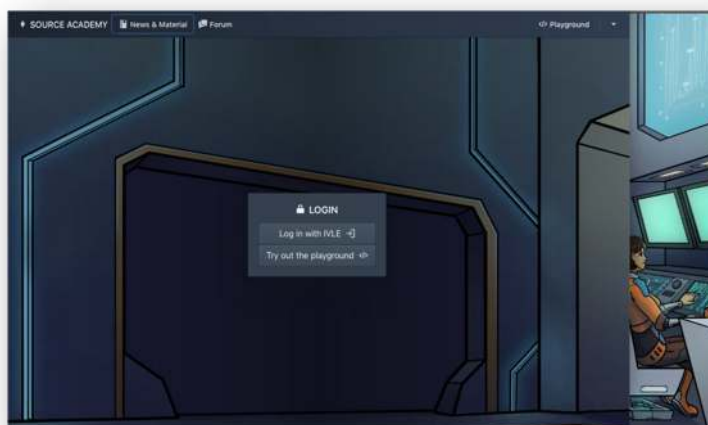


FIGURE 13 – THE START SCREEN FOR THE SOURCE ACADEMY

of the Source Academy was hosted on Amazon AWS servers and the source code was in a repository. During the execution of the course the previous students and current senior avengers were responsible for updating the source code and setting the various homework missions as well as fixing any bugs that were found. In short, the CS1101S students (and even staff) spent the majority of their time and effort working in the Source Academy.

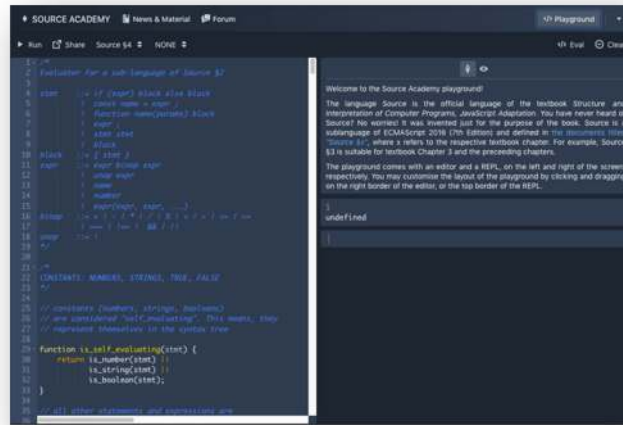


FIGURE 14 - AN EXAMPLE OF THE IDE IN SOURCE ACADEMY

Bells and Whistles

I use the phrase bells and whistles to categorize all the extra activities that took place in the CS1101S module that I personally deem extraordinary for the first programming course at a university. As previously stated, the CS1101S course had quests (extra programming assignments that were voluntary) that gave an extra module credit, game-based learning, and competitions to in sound, image manipulation and even a sumobot contest where the students programmed their sumobots to “fight”



FIGURE 15 - A SUMOBOT MATCH

other students’ robots and push them off the mats. Prizes were awarded to the winners in the different contests and the entire finale even had catering. Additionally, all the teaching assistants and staff had CS1101S t-shirts that we all wore during the duration of the course and those students who finished the course and took the final exam even received a small pin to be worn to show that they had completed the CS1101S module. In summary, no detail was missed, and every effort was made in the CS1101S module to make the module as exciting, challenging and fun as possible while also making the students and staff feel special and recognized.

CS1010J

The CS1010J module was a course in imperative programming at the Department of Information Systems that used the Java programming language. As previously mentioned, I joined the course a couple of weeks late, but the responsible professor Dr. Zhou Lifeng was more than accommodating to make it possible for me to contribute and collaborate in his course. My chief responsibilities in the course were setting and checking the midterm and final examinations. I attended all the lectures and also gave a lecture at the end of the module.

Instruction and pedagogy

The content and pace of the CS1010J module was basically identical to programming courses we have at University West. However, the instructional activities were every similar to the CS1010S module. Each week there was a lecture that introduced a new topic and then a reflection (called recitation in the CS1010J module) and studio that delved deeper into the weekly concepts and programming problems. However, the reflections were conducted by teaching assistants and not academic staff. The assessments in the CS1010J consisted of a practical programming exam, an MCQ midterm and a final that was a mix of MCQ and programming questions that required the students to write code by hand. The final grade was a weighted average of the assessments as well as the homework and attendance in the studios and reflections. In contrast to the CS1101S module, this the CS1010J module had a “standard” pedagogy in that the academic staff lectured, the teaching assistants



FIGURE 16 - ME GIVING A LECTURE IN CS1010J

provided support for problem solving and the students did their homework, practiced programming and took the assessments. There were none of the extras with gamification or contests for example and there was not an explicit focus on building learning experiences. However, I do not wish to portray this course in a bad light but rather highlight the extraordinary aspirations of the CS1101S module. The quality, ambition and effort of the students and staff was of a very high quality.

ICT Tools

The CS1010J module used ICT, of course, but it was more of a normal amount for an introductory programming course compared to the CS1101S module. Communication and resource sharing in the course took place through the aforementioned learning management system IVLE. The two other ICT tools of interest were CodeCrunch and DrJava. CodeCrunch is an online tool for completing and handing in programming assignments developed by NUS. It was used for students to write and hand-in programming homework assignments for grading. The CodeCrunch software automatically graded and provided feedback to the students. DrJava is a simple integrated development environment (IDE) for the Java programming language. It is typical in programming courses to suggest a specific IDE for the students to work with.

IT governance

Additionally, I shadowed the IT governance course given by the Department of Information Systems. I thought it would be interesting to see how a non-technical information systems course was given at NUS compared to my home institution. The IT Governance module basically dealt with teaching students how to be Chief Information Officers (CIOs) and manage an IT department. The course was led by an incredibly qualified and well-merited Professor Alex Siow. The course had a standard pedagogy that consisted of

lectures, tutorials and readings. The assessments consisted of a class test (20% of the final grade), group project (40% of the final grade), and final exam (40% of the final grade). The pedagogical interventions were weekly lectures and tutorials. The lectures were given exclusively by Professor Siow and a teaching assistant (PhD student) conducted the tutorials. The basic structure and pedagogy for these types of theoretical, information systems courses were very comparable to how we give similar courses as University West.

Computer clubs

Conclusively, two other interesting organizations that I encountered during my teaching sabbatical at NUS was the NUS Hackers (<https://www.nushackers.org/>) and the NUS Computing Club (<https://nuscomputing.com/>). The NUS Hackers is run by student volunteers and has weekly meetings every Friday called “Friday Hacks” and runs a series of technical workshops called “Hackerschool”. The goal of NUS Hackers is to make programming fun and build a healthy community of passionate programmers. This organization also provides a meeting place to network with other students, support to innovate and develop ideas and backing to maintain projects and even develop start-ups. The NUS Computing Club is a more formal organization that arranges a variety of student activities such as mock practice exams, homecoming, exam welfare packs, fun runs, etc. Both of these organizations provide further pedagogical support and guidance for all things computer and programming related and complements the formal education. Also, they help develop and promote a culture of learning and innovation.

Pedagogical center and courses

In addition to my actual teaching workload described above, I took advantage of the immense pedagogical resources offered by NUS. NUS has a Center for Development of Teaching and Learning (CDTL, <http://www.cdtl.nus.edu.sg/>), which has a vision to innovate and share good practices in teaching and learning and in the use of technology and emerging media for creating significant learning experiences. It is a campus wide institution that offers courses, seminars, workshops, and support so that teachers can deepen their pedagogical knowledge and share practices and ideas to further strengthen the quality of teaching as NUS. In short, it is a fantastic organization that shows how seriously NUS approaches teaching and learning by offering a number of interesting pedagogical workshops and conferences each semester as well as numerous digital pedagogical resources and tools that are available for free. During my stay at NUS I was able to participate in three workshops and two conferences and I tried to partake of as much of CDTL resources as possible.

The first and primary workshop that I participated in entitled *Developing a Teaching Portfolio* took place in three separate, 3-hour sessions spread throughout the semester. It was very serendipitous that this workshop was available because a proper teaching portfolio will be necessary to further my academic career at University West. The workshop provided me with invaluable practical information and insights on how to best structure and present my teaching philosophy and practices in a portfolio.

The workshop was led by Associate Professor Johan Geertsema (who is also the director of CDTL) and Dr. Mark Gan, who were both very knowledgeable and did a fantastic job of encouraging the participants to

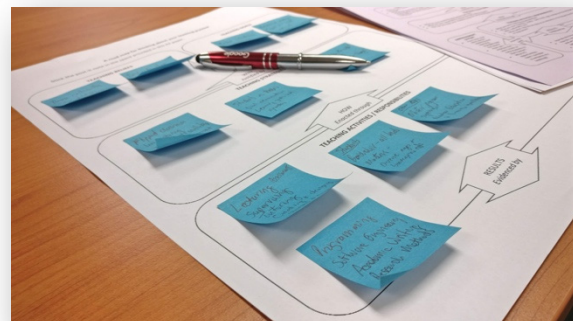


FIGURE 17 - SOME RESULTS FROM THE TEACHING PORTFOLIO WORKSHOP

actively contribute and reflect on their own teaching beliefs and practices. The writing of my teaching portfolio will definitely be a concrete result from my time at NUS.

Additionally, I managed to participate in two other workshops: *Online Synchronous Discussions for Tutorials* and *Designing an Authentic Learning Environment*. I was rather familiar with online synchronous discussions and tools, but it was a nice refresher and the course was conducted online using the software Zoom in order to accentuate learning and using a tool for synchronous discussions. The workshop for designing an authentic learning environment was also very informative and well-done and somewhat of a revelation for me. I was acquainted with the term but did not have a good grasp of exactly what the idea of authentic learning entailed. The workshop provided good examples of authentic learning as well as a framework as to how I could implement all or aspects of authentic learning in my own teaching. It became apparent that many aspects of authentic learning overlap with work-integrated learning which is a prioritized research area at University West, so the topic of authentic learning is something that I plan to explore further and possibly implement in my teaching in the future.

Finally, there were also two pedagogical conferences that took place during my stay at NUS. The first was the *8th Teaching and Learning in Higher Education Conference (TLHE 2018)* that was given during the first recess week of the semester. *TLHE 2018* was a conference that focused strictly on teaching and learning in higher education. As an interesting way to present, the conference used *PechaKucha*, which is a Japanese concept of using a maximum of 20 images with 20 seconds per slide (20×20) as the presentation technique. The images progress on a timer and the presenter narrates them as they go. The first half of the day was filled with two keynotes about using research as an instructional tool from the very start of higher studies. The second half of the day consisted of breakout sessions on various pedagogical topics. Some interesting thoughts from the keynote speakers were for example that the course/module-based structure of higher education is like stamp collecting for students. They collect small pieces of knowledge, but never make the connections or see the big picture of a subject or research area. This idea was mentioned by Dr. Dilly Fung (keynote speaker) and you can read all about it in her free book. Another example from New Zealand and ecology was that they strictly use research as an instructional tool and six students have managed to publish in scientific journals before finishing their undergraduate degrees. In summary the conference was very worthwhile.

The second pedagogical conference was more informal and similar to a *Ted Talk* and it was entitled *Learning on NUS Campus 2018*. This conference covered a wide variety of topics and was given in a number of parallel sessions throughout one day and was a new initiative by Provost Professor Ho Teck Hua for faculty members to share their subject area in an accessible and fun way to audiences from diverse backgrounds. I attended a number of sessions and found them interesting both from a content and pedagogical perspective. The first session I listened to was *AI Overlord: Should we be very afraid?* It was an enjoyable presentation that explained very simply how AI is just software that is not rule-based/algorithm-based like standard software; in other words, non-AI software is limited by the rules we humans provide. The presenter did a great job of explaining neural networks and deep learning and of course he finished with the Google Alpha Go success story. Another session I attended discussed the Internet in general with a question about what the Internet is for them in his presentation *Learning about the life and lore of the Internet*. Two answers were leisure and social connectivity. He basically summarized his experience with a course about the Internet as a phenomenon and how students use and view the Internet. Interesting and intriguing to see how his students perceive the Internet. Lastly, the final presentation of the day was about live coding and moving from a pedagogy of showing static code examples to using live coding in programming courses. This is something

that we generally do at University West and definitely at NUS. My reflection is that this is something we should try to integrate into the programming courses we are giving to middle and high school teachers in Sweden. The tool the presenter showed was jsfiddle.net, which is an online collaborative coding environment for JavaScript. The presenter even explained the PRIMM method and the micro:bit controller used in the UK to teach programming, and these tools could also be useful to experiment with in the future.

Reflections about pedagogy

After five months of working at SoC at NUS, I have started to form some introspective opinions regarding the pedagogy and instructional activities that take place at NUS and Sweden. Firstly, in my case a very large and glaring difference was scale and resources. The sheer number of staff and students in a module was astounding and I have probably not interacted with so many different students over that past 10 years. Our Bachelor's program at University West typically has around 50 students each year. Furthermore, the availability of staff and teaching assistants made it much easier to address pedagogical issues by simply offering extra lectures or reflections, which is something we would seldom be able to do in Sweden as we generally lack surplus manpower. Also, in Sweden I typically have 4-6 courses per semester but at NUS it seemed that the standard number of courses per semester was two.

Secondly, another conspicuous difference was my perception of the ambition and drive of NUS students and staff. I deem Swedish academic staff and students as ambitious and driven, of course, but our Singaporean counterparts raised the bar to another level. Regarding the academic staff, every detail and aspect of a course was well-planned and executed and nothing was left to chance. As soon as there was a problem or issue, it was addressed. Everyone always did their job at a high-level. Regarding the students, they were also extremely motivated and focused on obtaining the highest grade possible. I frequently saw students studying late on weekends and it seemed that it was more of a cultural norm that you studied all the time instead of "partying" on the weekends. This difference can be cultural or due to the fact that NUS is a highly ranked educational institution and thus attracts the most ambitious and capable students from all of Southeast Asia. One downside to this ambition is that there was a general focus on grades that does not generally exist in Sweden. A risk that can arise from grades becoming the focal point in a course is that learning can suffer as the goal moves from learning to instead getting a high grade.

Thirdly, NUS keenly emphasizes teaching and learning. This emphasis is not a vision or intention, but instead something that is carried out in everyday practice. There is a focus on building rich learning experiences for all the students i.e. extra challenges for the most ambitious and gifted students and a constant tracking of struggling students with appropriate supplementary pedagogical interventions. This focus complements the emphasis on learning and not teaching. The staff, especially in the CS1101S module, have designed the module to minimize the authoritarian "teacher knows best" mentality and students are encouraged to actively engage with problems and concepts themselves with the teachers as mentors and guides instead.

Fourthly, two specific instructional interventions that I found useful and intend to explore further were reflections and MCQs. Both measures are not revolutionary on their own, but I believe they are techniques that are underused at my home institution. The careful mixture of lecturing and active student participation to create a learning experience in a formal and structured way as reflections are used at NUS, I believe can improve the learning outcomes in our programming courses at University West. In the same vein, MCQs are a very common form of assessment, but one that has fallen out of favor for some reason. I now deem that a

properly designed and executed MCQ can also help improve learning outcomes in our programming courses in Sweden.

Fifthly, another key consideration from my time at NUS is that making challenging courses fun and using aspects of gamification and game-based learning to highlight the fun aspects of a difficult subject presents a viable means to achieve good learning outcomes. The aforementioned game-based Source Academy, the gamification and collection of experience points for quests, and all the other bells and whistles helped alleviate the mundane and tedious hours of studying and writing code to cultivate the skill to create algorithms and write good programming code. Once again this is an area that is underdeveloped at University West and worth exploring further. Making a course at university fun does not necessarily mean that the course becomes less serious or trivial.

Finally, on a larger pedagogical scale, I feel strongly that University West can improve the quality of teaching and learning by creating their own version of the aforementioned CDTL. A centralized organization that specifically focuses on pedagogical practices across all disciplines and that offers workshops, seminars and other resources to support teaching and learning can be an invaluable addition to the quality of all the educational programs offered at University West. This proposed organization can even provide a formal forum for the sharing and dissemination of best practices for teaching and learning that currently take place by word-of-mouth in separate departments and institutions instead of throughout the entire university.

Action Plan for Teaching and Learning

As previously mentioned, the pedagogy at NUS was not unexpectedly innovative or revolutionary, but instead focused and willing to experiment and test a variety of existing tools and pedagogical interventions. My plan is to adopt this spirit of openness to investigation and change into my own teaching. As a step in establishing a clear strategy as to how to summarize all of my experiences and reflections from my teaching sabbatical at NUS, I propose the following items in a teaching and learning action plan. The first list of proposed action items consists of interventions and changes that I have a great deal of control over, thus the likelihood of implementation is higher. The second list contains things that I would like to see implemented but have less control over. The following ideas are pedagogical changes that I plan to implement and test in my own teaching at University West in some fashion during the coming year:

1. Develop and use multiple choice questions to test factual knowledge in a variety of courses.
2. Use reflections (the hybrid lecture/lab intervention) in our programming courses.
3. Experiment with using assessments that require handwritten programming code in our programming courses.
4. Formalize and improve the process of helping struggling students.
5. Share my pedagogical experiences with colleagues and encourage innovative pedagogy.
6. Write my teaching portfolio.

The following ideas are pedagogical changes that I would like to see implemented at University West in the future:

1. Have University West establish their own center for teaching and learning, similar to the CDTL at NUS.
2. Establish and promote a culture of experimentation to design learning experiences.
3. Create learning experiences in all our programs that cater to all students, both strong and weak.

Visit from University West

As promised by the Vice-Chancellor of University West, Professor Martin Hellström and a small delegation from the Department of Informatics visited both NUS and Nanyang Technological University (NTU) in the beginning of November. My fellow STINT teaching sabbatical recipient Marcia Håkansson Lindqvist, who was doing her teaching sabbatical at NTU, was very kind to host and guide the delegation from University West and arrange a visit to the educational and pedagogical resources, structures, and institutions at NTU. The objective of the Vice-Chancellor and delegation at University West was to see firsthand my experiences at NUS as well as explore and seek inspiration in the pedagogical and educational infrastructures and resources used at both NTU and NUS. The visit was short but intense and hopefully their experiences will help further pedagogical change at my home institution. I highly recommend that future STINT teaching sabbatical recipients arrange visits from their home institutions to increase the probability of implementing pedagogical changes.



FIGURE 18 - THE DELEGATION AT UNIVERSITY WEST VISITING NTU AND NUS

STINT Midterm Seminar

As part of the STINT teaching sabbatical, the recipients in the US and Asia gathered on their respective continents to discuss the progress of their individual teaching sabbaticals in order to discuss and share experiences. The 2018 midterm seminar for the Asian universities took place at the Chinese University of Hong Kong. The midterm seminar was a great opportunity to hear the experiences of the other recipients and compare and contrast their experiences with my own. It also provided an opportunity for me to summarize my own experiences and reflections.



FIGURE 19 - ALL THE STINT TEACHING SABBATICAL RECIPIENTS IN ASIA

Life in Singapore

Obviously, my experiences and reflections of working at NUS are the focus of this report but living in Singapore with your family and experiencing a new country and culture are also worth mentioning briefly. Singapore is a fantastic, multicultural country that offers everything from tropical beaches, jungles and wildlife to luxury shopping malls and skyscrapers. It is an extremely safe and organized society that presents an interesting mix of Western and Asian cultures. On the weekends and holidays make sure that you take advantage of the international cuisine and hawker food culture, see all the popular tourist attractions like Gardens by the Bay, Merlion, Marina Bay Sands and Sentosa Island and enjoy the tropical weather (always hot and humid and almost daily rain).



FIGURE 20 - THE MARINA BAY SANDS HOTEL AND GARDENS BY THE BAY AT NIGHT

Concluding Remarks

The STINT teaching sabbatical is an incredible opportunity both professionally and personally. Professionally, it offers the recipient the opportunity to work at a world-leading academic institution. Personally, it offers an opportunity for the recipient (and his/her family) to experience another culture and country. It is a challenge and adventure that I hope all Swedish academics apply for at some time in their careers. Even if an individual recipient's experiences are less than expected or desired, the teaching sabbatical offers the possibility to break one's own teaching routine and pedagogical habits and observe and test new pedagogical attitudes and interventions. The STINT teaching sabbatical provides an ideal context to experience new teaching and learning practices abroad and hopefully integrate them into higher education institutions in Sweden. And last but not least, I would also like to express my sincere thanks to the STINT foundation for providing me with this opportunity.

References

Spence, L. D. (2001). The Case Against Teaching. *Change: The Magazine of Higher Learning*, 33(6), 10–19.

<https://doi.org/10.1080/00091380109601822>