The Computer Science Student's handbook to First Year at UTSC

Written by a very bored Kohilan Mohanarajan

July 14, 2017

Contents

1 Foreword

If you're reading this, you are most likely a first year student entering the University of Toronto Scarborough's computer science program, in which case, welcome to UTSC! These next four, maybe even five or more years will be filled with ups and downs around every corner.

Perhaps the biggest fear a first-year computer science student faces entering University is the possibility of being unprepared. As such, out of pure boredom on a Sunday afternoon, I've decided to create this little handbook to help you guide you through this eventful first year of your life at UTSC.

ATTENTION: This document shouldn't be taken as gospel, but as a supportive resource to help you with your experience. At the end of the day, your experience is yours only, and shouldn't be dictated by what someone else has told you. University and Academia in general is an experience that not everyone gets, so make the most out of your time here.

2 Getting Around: The Basics

At UTSC, there's two ways to get around: On Campus, and on the Web. In this section, we'll be delving through both.

2.1 The Campus

UTSC is a relatively small campus, so getting around isn't too bad. With that being said, here's a few key places you should get familiar with:

- IC130: Located in the Instructional Centre Building, you're bound to have at least one lecture in this room during the year, as well as a couple of exams
- HW216: Located in the Humanities Wing, this room is another lecture hall you'll find yourself becoming familiar with throughout the year
- The rest: AA112, SY110: Located in the Arts and Administration Building and the Science Research Wing respectively, more big rooms that you'll encounter at some point.
- AC223: While it's highly unlikely you're going to have any lectures here, you are likely to have at least one exam in this room, located in the Academic Resource Centre, this year, as it's one of the biggest lecture halls at UTSC, if not the biggest.
- The food spots: UTSC doesn't really have much in terms of food, to be frank, but what it does have is...it's alright, I guess.

There's a Tim Horton's on the first floor of the Bladen Wing, which is where you'll find yourself before your morning lectures standing in line for 20 or so minutes while the line, which extends out of the actual restaurant, moves at a snail's pace. There's a smaller Kiosk offering more barebones offerings (Tea, Coffee, donuts, muffins), but with the recent strike, it might be a while before that opens up again.

The Marketplace can be found in the hallway leading to the Humanities Wing, and consists of small shops like a Pizza Pizza kiosk, a Booster Juice, a Spring Roll, as well as a couple of other food options. Food here usually will run you between five to ten dollars, and offers a decent variety of options.

The Student Centre houses five restaurants, which offer another range of options for those who don't want to shell money out at the Marketplace.

There's KFC Express, which offers a stripped down KFC menu (which ironically doesn't have Chicken drumsticks or thighs). The favorite pick is the snack box, which gives a pretty good bang for your buck for when you're racing between classes. Wait times here are actually relatively good, ranging between 3-5 minutes. To date though, you may find that the fountain drinks are a bit flat.

Hero Burger hosts more options akin to its regular chain restaurant, but you'll find yourself waiting for a long while to get your food. The drinks here are fizzier than KFC, though.

Subway also offers the usual selection, though promotions aren't carried, in case you were looking for that limited-offer habanero sauce sub. Lines are relatively quick, like your average Subway restaurant.

Perhaps the newest of the five restaurants, Treats has a pretty good selection which'll run you between five to ten dollars, so if you're craving a pita wrap fix, go here.

Finally, there's Asian Gourmet. This place fluctuates in quality very often, and will run you a pretty penny too. Food tends to range between average to dry and bland, but if you're in desperate need of some General Tso's Chicken, then this'll have to be your choice.

Outside of the Student Centre, you'll find Nasir Al-Huttam with his hot dog stand. This food stand is a **MUST** if you're studying at UTSC. You have not truly gone to UTSC unless you've had food from Nasir's food stand. He has everything from veggie dogs to Italian Sausages, fries, poutine and more! Top notch quality that'll run you, at max, 7$.

- The off-campus food offerings: If you're dead tired of the food at school, unfortunately, there isn't much outside, unless you're willing to make a trek. In a plaza next to the Toronto Pan-Am Sports Centre, you'll find a Popeye's,

a Reginos, Panam's Joint, the local bar, and Osmow's. Osmow's is probably the best of the bunch, giving you the best bang for your buck. If you're daring enough, try the Suicide Shawarma.

- IC308, The AMACSS Office: If you're looking for some help with your Academics, this is where you'll be able to find your local AMACSS exec, who'll be able to help you. There'll be more on AMACSS later on in the handbook.
- IC third/fourth floor: This is where you'll find the various offices of your Professors and Lecturers. At the time of writing this handbook, only Brian Harrington, Thierry Sans, and Nick Cheng's are on the third floor.

2.2 The Web

- Blackboard: located at www.portal.utoronto.ca, this'll be your homebase for most of your math courses, as well as electives. Computer Science courses rarely use Blackboard, instead going for their own website, usually located at a website like www.utsc.utoronto.ca/ instructorname/coursecode.
- ACORN: Another huge site you'll find yourself logging into often, this is where you'll be choosing your courses and programs, paying for said courses and programs, and receive your marks for your courses.
- Piazza: This is the class forum used by all of your CS courses. This is usually where Instructors make their announcements, but the main point of Piazza is to offer a person-to-person helpline for exercises and assignments, be it instructors helping students, or students helping students. Piazza experiences usually range from very helpful to downright mind-bogglingly inane. Have fun!
- Markus: This will be where you submit your exercises and assignments for your CS courses. It's a simple process, which requires you to upload the necessary .py or .pdf files needed. Keep in mind to name your files properly, it might end up biting you back if you don't.

3 Mandatory Courses

Like every program, Computer Science isn't without its mandatory courses. These following six courses will be taken by each and every First Year student, and end up contributing towards your application towards POSt (Which we'll talk about later on in the handbook) Textbooks for the courses that have them (The math courses, basically) will be linked to in the "Useful Links" section of this handbook.

NOTE: The content of these courses change from year to year, so there may be some slight discrepancies here and there based on who's teaching or what content's been moved around.

3.1 CSCA08: Introduction to Computer Science I

UTSC Calendar Page

This is the very first CS course you'll be taking at UTSC, usually in your Fall semester. This course introduces you to the fundamentals of your average high-level programming language, taught through Python. You'll learn basic concepts such as iterative statements, loops, functions, and Object Oriented Programming (OOP) in this course.

In previous years, there have been three major assignments, 5-6 quizzes, as well as 10-11 smaller exercises. The exercises are released at the end of every week, usually expected to be completed by the end of the following week. The exercises are usually released on the instructor's respective course website, and are submitted through Markus, our proprietary submission website. Exercise content consists of making functions, or writing a specific bit of code, which gets run through an automarker, which assigns a grade based on how many of the test cases your program passes.

Quizzes are usually handed out at random during your tutorial, and will cover the topic discussed in class the week before. Quizzes are also really short, being only a page long, and don't require you to do *too* much thinking.

Assignments are some of the biggest pieces of coursework you'll be doing in this course. They encapsulate multiple weeks' worth of content, and require you to show your understanding of said concepts through the code you write. Assignments are not only automarked to check for whether or not it will always run properly, but also marked by a TA for documentation and coding style. The assignments also grow in difficulty, with the last one being the hardest of the three.

There are also two Term tests in place of one midterm examination, mainly to give you more chances to do well, in case you don't do well on one of the two examinations. These tests consist of questions that deal with tracing through given code, writing code given a set of instructions, and Code Mangler questions. (Oh, you'll *love* Code Mangler.)

Key Advice: Stay Diligent, get your work done on time, and start the assignments as soon as they're handed out, and you should make it through this course comfortably.

3.2 CSCA48: Introduction to Computer Science II

UTSC Calendar Page

The second CS course you'll take at UTSC, this course expects you to have a strong understanding of the concepts you learned in CSCA08, as the course content begins to deviate from teaching programming to teaching concepts of computer science, such as Data Structures, Recursion, and Complexity.

Like A08, this course has 10 exercises, and 5 quizzes, which take around the same amount of effort as they did in A08, albeit with more complicated material. There are two big assignments this time around, which serve the same function as they did in A08.

The term tests are also more difficult in comparison to A08, so don't let your guard down when writing them.

Key Advice: Like before, stay diligent, do your work on time, and in preparation for your Term Tests, work through the supplementary questions given out in Practicals, or try and devise questions of your own. Questions can come out of the blue in this course.

3.3 CSCA67: Discrete Mathematics

UTSC Calendar Page

This, unlike the other two CS Courses, is the most unlike Computer Science at first glance, but serves a great importance to your studies moving into subsequent years. This course has been deemed by some as the indicator as to how far one will go with their studies in Computer Science, and with good reason.

You may be asking yourself what this course even consists of, and to answer that, we must divide the course into two halves: Proofs, and Counting.

In the proofs section of the course, you'll be learning the basics to writing formal proofs. This will include content such as truth tables, predicates and quantifiers, modulus proofs, and induction. Induction will be the most important concept from this part of the course, as it carries on to not only second-year courses, but also MATA37, hence why this course is one of MATA37's prerequisites.

In the counting section of the course, you'll learn basic combinatorics, such as Combinations, Permutations, and Selections, as well as going over basic Probability, like Bayes Theorem. This is considered to be the easier portion of the course, but don't take that for granted.

CSCA67 consists of 8 exercises, each of which consists of roughly 8-10 questions, given out each week, to be completed by the following week, as well as two big assignments, which contain roughly 7 questions, albeit harder than the exercises. The assignments are due two to three weeks after they've been handed out. The exercises, as well as the assignments, are weighted more than the midterm, which makes sense, as the class average for the midterm is always very low. Most students usually get caught up on the midterm, but don't let that dishearten you.

Key Advice: Keep up with the course content, as the exercises for this course are usually due the same day as the exercises for CSCA08. Practice makes perfect, and strong knowledge of the concepts can serve as a basis for doing well on Assignments and the midterm.

3.4 MATA31: Calculus I for Mathematical Sciences

UTSC Calendar Page

This is the first of the three math courses you'll take in first year, and is your first exposure into theoretical maths. Based on how they structure the course, it's gonna be 60/40 theory to computation. You'll learn more about writing formal proofs in this course as well, although not to the same extent as CSCA67.

Practice and memorization will help you get along easy in this course. There are five assignments and five quizzes, each alternating from week to week. The assignments are long and tedious, but serve as fair practice of the material. Material-wise, the first six weeks will be new content to many, while the latter six weeks will be familiar content.

Key Advice: Practice and memorize, become familiar with the content, as this is the easier of the two calculus courses this year.

3.5 MATA37: Calculus II for Mathematical Sciences

UTSC Calendar Page

This is the second Calculus course you'll take in first year, and by far, the hardest. This is considered by many to be a GPA killer, and usually ends up being the bane of most people's POSt applications. It is a continuation of Calculus 1, but the vast majority of this course will be new to you if you didn't take AP Calc in High School.

This course deals with Reimann Sums, Integration, Sequences, and Series from a very theoretical aspect. Like A31, there are 5 quizzes and 10 assignments, however this time, the order in which you hand in assignments and do quizzes is randomized, meaning that as you walk into your tutorial each week, you won't know whether you hand in the assignment you did over the past week, or if you're writing a quiz based on the material from the assignment. This forces you to complete every assignment and learn the material, and learn the material you shall if you want to succeed in this course.

The midterm and final exam are comprised of questions that not only resemble those you've seen, but test your knowledge of the material with new question types, and questions you may have never seen prior. Good luck to you as you go through this course, you will need it.

Key Advice: STUDY. HARD. Don't slack off, or fall behind with the course material, as it will only harm you further. True mastery and understanding of the content is necessary if you want to do well in this course.

3.6 MATA22: Linear Algebra I for Mathematical Sciences

UTSC Calendar Page

Linear Algebra is arguably the easier of the two maths you will be taking in your winter semester, however, that may or may not change this year, as the 2017-2018 year marks the first year that Computer Science students take MATA22 instead of MATA23. The change towards making MATA22 a Mathematical Science-focused course will mean more emphasis on the proofs and theory of the course.

This is a course where the textbook will actually be very useful; in fact, the textbook is perhaps your best resource for this course, so use it to the best of your ability. In this course, you will go over Cartesian Linear Algebra, Matrices, Determinants, Vector Spaces, Eigenvalues, and Eigenvectors, amongst other concepts.

While this will be the easier of the two math courses in the winter semester, do not slack with the material given. In MATA23, there were 11 non-mandatory assignments to practice the course material, as well as five quizzes, given every two weeks. As mentioned before, this may change with the introduction of MATA22.

Key Advice: If you want to do well in this course, do the assignments, as questions for the quizzes are ripped from them, and just get a good understanding of the theoretical aspect too.

4 Electives: Birds, Breadths, and Bamboozles

As with any other University Program, students are required to take courses that don't pertain to their program, perhaps to broaden the scope of their learning experience. Obviously, this would lead to many seeking the easy way out, which is to say, finding the easiest courses that can help them fill their necessary requirements. The following list of courses shouldn't be taken as the final word, as mentioned above, course structuring can vary from year to year, sometimes even semester to semester. Do your research before diving into this $640+ investment.

4.1 Some notes

1. You don't *need* to finish all your breadth courses by first year. If you can't take a course because of scheduling conflicts or whatnot, then don't fret it. You just need to complete them before you graduate.

2. You cannot take 1 credit (two courses) that satisfy one breadth and replace another breadth. That is to say, you can't take two Arts, Literature, and Language courses, and have the second one replace your History, Philosophy, and Cultural Studies breadth.

3. You can CR/NCR a breadth requirement course, and have it still count towards your degree.
4. On top of your Breadth requirements, you also need to fulfill a writing requirement as well before you graduate. For your convenience, each course has been labelled as to whether or not it satisfies the writing requirement.
5. Unlike breadth requirements, if you CR/NCR a writing requirement course, it won't count towards your degree. It will count towards your breadth requirement, but not your writing requirement.

4.2 Arts, Literature, and Language

4.2.1 ENGB02: Effective Writing in the Sciences

UTSC Calendar Page

A very easy course, which consists of light coursework, an essay, a midterm, and a final. The course exists to teach science students how to write proper. Worth taking if you're looking to improve your writing skills. Space is limited for this course, so keep that in mind.

Bird or Bamboozle?: Bird for sure.

Does it satisfy the writing requirement?: No

4.2.2 VPMA95: Elementary Musicianship

UTSC Calendar Page

If you've got any experience whatsoever in music theory and/or have read sheet music before, this course is a breeze. The course assumes you have no formal knowledge of music theory whatsoever, so you'll be starting off with the basics, like notes on a staff, major/minor scales, intervals, circle of fifths, etc. Overall a very easy course, and a very interesting course too.

Bird or Bamboozle?: Bird for sure.

Does it satisfy the writing requirement?: No

4.2.3 LINA01: Introduction to Linguistics

UTSC Calendar Page

This is like VPMA93, though some could argue the content is easier than the former. Again, if you've got great memorization skills, go for this course.

Bird or Bamboozle?: This one's a bit on the fence, as you may or may not find it easy.

Does it satisfy the writing requirement?: Yes

4.3 History, Philosophy, and Cultural Studies

4.3.1 VPMA93: Listening to Music

UTSC Calendar Page

This one is a deceiver. While the content of this course could be deemed as easy, there is a lot of material to memorize. To memorize all the material in this course, and still have time to do well in your other courses, you'll need to have good memorization skills.

Bird or Bamboozle?: Issa Bamboozle, folks.

Does it satisfy the writing requirement?: No

4.3.2 PHLA10: Reason and Truth, and PHLA11: Introduction to Ethics

UTSC Calendar Page (PHLA10)
UTSC Calendar Page (PHLA11)

Both of these courses fulfill a writing requirement as well, and are possibly two of the easiest ways to get it. If you've got an open mind, and can sustain an argument well, then both of these courses will be cakewalks for you.

PHLA10 deals with the common-perceived side of Philosophy, asking the hard-hitting questions about our existence and religion. PHLA11, on the other hand, talks about the philosophy of ethics, through the eyes of both Kantian Contractualism and Welfare Consequentialism. They'll really get you thinking too, so the interest factor is also high in these courses. Definitely take the time to take these courses if you can.

Bird or Bamboozle?: Bird.

Does it satisfy the writing requirement?: Yes

4.4 Social and Behavioural Sciences

4.4.1 SOCA03: Introduction to Sociology

UTSC Calendar Page

Another case of VPMA93 syndrome, this is a course that requires you to memorize a lot of stuff, but there are other things like essays and journal entries to soften the blow along the way. Like the Philosophy courses, this course will require you to think outside of the box. Do note that this is the first year that SOCA03 is being offered, taking the place of both SOCA01 and SOCA02. This means that this course will span an entire year rather than just one semester.

Bird or Bamboozle?: If you can put in the time, this can be a bird course for you.

Does it satisfy the writing requirement?: No

4.4.2 MGTA01/MGTA02: Introduction to Canadian Business

UTSC Calendar Page (MGTA01)
UTSC Calendar Page (MGTA02)

Considered to be some of the easiest courses at UTSC, these ones are just pure memorization. A02 is a bit harder than A01, but the material in general is very simple and intuitive to memorize, making it less of a burden than something like VPMA93.

Bird or Bamboozle?: Bird.

Does it satisfy the writing requirement?: No

4.4.3 CSCD03: Social Impact of Information Technology

UTSC Calendar Page

One of the few, if not only, Computer Science courses that fulfill a breadth that isn't Quantitative Reasoning, CSCD03 is a course that all Computer Science students need to take before they graduate. If you don't want to have to take a Social Science breadth, just wait until fourth year and take this course and satisfy the

Social and Behavioural Sciences Breadth.

Bird or Bamboozle?: Apparently a bird.

Does it satisfy the writing requirement?: No

4.5 Natural Sciences

4.5.1 EESA06: Introduction to Planet Earth

UTSC Calendar Page

Another case of VPMA93 syndrome, just pure memorization. Depending on when you take the course, you might also have a poster project worth 40% of your mark, so that alleviates things. There's a lot of test banks and resources out there that make this course a bunch easier, so resource-wise, this course is a gold mine.

Bird or Bamboozle?: Bird, but you'll have to memorize a lot.

Does it satisfy the writing requirement?: No

4.5.2 EESA09: Wind

UTSC Calendar Page

Yes, the course's name really is, "Wind". Don't take it. Out of all the courses on this list, this would be the one that I personally would recommend the least. Actually, if you're willing to spend one C/NCR credit on a course, let it be this one. This has got to be some of the most mundane, boring material that I've ever had to sit through and study. To make matters worse, the amount of material to remember is also insane. There are weekly quizzes on blackboard, but unlike other courses like MGTA01, the questions are all randomized, so it works against your favor. The lectures, which are 3 hours long, will feel like the longest three hours of your life.

Bird or Bamboozle?: BAMBOOZLE. If you're a masochist, or want to experience masochism, take this course. Otherwise, steer clear of it.

Does it satisfy the writing requirement?: No

4.5.3 ANTA01: Introduction to Anthropology: Becoming Human

UTSC Calendar Page

Considered to be a very easy course, ANTA01 consists of 10 % just for showing up to tutorials, and a 1500-2000 word essay that they give a month to complete. It

will involved a bit of memorization, so do keep that in mind.

Bird or Bamboozle?: Bird.

Does it satisfy the writing requirement?: Yes

4.6 Quantitative Reasoning

You're in Computer Science. This one is a freebie.

5 Co-op at UTSC

If you're a computer science student at UTSC, you're more than likely also enrolled in the co-op program. You may have many questions about the co-op program here, and what the experience consists of, and this paragraph will do its best to give you a brief glimpse into this experience.

Before your first work term, you'll need to take two prep courses, and one work term search course. COPD01 is the first course you take, usually in the fall of your first year if you were admitted into coop computer science, or in the summer, if you opt to start your coop journey in second year. This course is...somewhat bland, to be quite honest. It's made to serve as a precursor of sorts to the subsequent courses. You'll be learning some skills like writing a resume and writing a cover letter. If it's offered online, I'd recommend taking it online.

COPD03 is the next course you take, usually in the winter of your first year, or the summer after. Which semester you get COPD03 scheduled for is up in the air, and at the discretion of the coop office. If you get COPD03 scheduled for the winter, that means your first work term will be the fall of that year. If you get COPD03 scheduled for summer, then your first work term is the winter of the next year. COPD03 is the most worthwhile co-op prep course, in my opinion. You'll be going more into depth of what applying for jobs is like. You'll also engage in mock interviews to help prepare you for your work term search.

COPD11 is the final course you'll take, during the semester before your work term. This is a generally bland course, but the bulk of the work is applying for a job. This is done through the proprietary CSM portal. You can also apply for jobs outside of CSM, though they need to meet certain criterion.

One thing that you need to keep in mind is that in order to maintain your placement in the coop program at UTSC, you need to maintain a 2.5 cGPA.

6 Social Life at UTSC: Yes, it exists.

One may think that Computer Science is a very Xenophobic discipline, and that's...I mean, I guess it holds true in certain facets, but that's besides the point. A large

part of doing well, but also having a fulfilling experience at UTSC is being involved in the community. Networking is a large part of Computer Science (The social kind, not the one that uses Sockets), and will be pivotal to even the courses you take.

Try and get yourself into a group chat, be it one for individual courses, or the CS program in general. This is a great way to meet new people, but also get more direct help. Who knows, someone in that group chat you barely talk in might have the answer to that burning question you've been having about some piece of coursework, and conversely, you, implying you are the benevolent reader I assume you to be, can give help to those who need it. Together, you can build strong bonds that will carry throughout subsequent years. (Unless of course people drop out. What? It happens.)

Another thing to do is to get involved with organizations, clubs and teams. These provide a release from your coursework, and can even teach you something new. Attending Hackathons are also great experiences, and can really help expand your network, while acquiring new skills and swag. If you do nothing but study, University life is bound to get boring quite quickly. At the end of the day, University is just as much a social experience as it is a learning experience, so make the best of your time here before it's gone.

7 The POSt Talk

While reading through this handbook, you may have encountered this term a couple of times by now, and you may be left wondering what the heck a POSt even is. When you're accepted to UTSC, you're accepted into the Computer Science, Mathematics, and Statistics Department. What this means is that you technically aren't in a Computer Science Major just yet.

A subject POSt (**P**rogram **O**f **St**udy) is what you apply to at the end of the year, and is the respective Major/Specialist/Minor that you go into after first year. For instance, you can choose to be a Computer Science Major after first year, alongside two minors, or do a double major in Computer Science and Management, alongside one minor. There are also specializations, such as the Software Engineering Stream or the Comprehensive Stream, which are more focused on specific parts of Computer Science. These are usually the Subject POSts most Computer Science students end up applying to.

However, unlike most other disciplines, these Subject POSts are limited, which means that certain criteria must be fulfilled in order to apply. As of April 2017, the current *guarantee* (This is important to note) to get a Subject POSt of your choosing is a 2.75 GPA or above across your six mandatory courses. This is a guarantee in that if you achieve this GPA, then you are automatically guaranteed a spot in your Subject POSt, no matter what. With that being said, the 2.75 GPA is not a hard cutoff, in that if you do not achieve that GPA across your core courses, you are not removed from the Computer Science program. Those who aren't able

to achieve the 2.75 GPA are then put into consideration, and from that, a certain amount of people are chosen to enter their Subject POSt.

So, while not achieving that 2.75 does not outright mean that you've been denied from your Subject POSt, it means that your chances of being accepted are at the mercy of the CMS department. You'll find POSt to be a huge point of contention amongst first years, as it's considered the ultimate goal for the end of the year. At the end of the day, while getting into your subject POSt is something that everyone should be striving for, don't let it be the end of your career in Computer Science if you don't get in. You are allowed to apply again after having taken your second year courses, in which those will also be taken into consideration alongside your six core courses from first year. Good luck to you, and may the odds ever be in your favour.

8 Computer Science Extracurriculars

Albeit being a relatively small campus, there are several clubs, teams, and organizations at UTSC. Of those clubs, there are currently two main ones that serve the Computer Science Community: AMACSS and CSEC. While these two are the CS-related clubs, feel free to try out for other teams and clubs as well!

8.1 AMACSS

AMACSS, or the Association of Mathematical and Computer Science Students, is UTSC's Main DSA (Departmental Student Association), representing students from the Mathematics, Statistics, and Computer Science Students. They host a variety of events throughout the year, such as mixers, gaming nights, and most importantly, Seminars.

They will host review seminars for most, if not all first year Computer Science, Mathematics, and Statistics courses before midterms and finals, which can be used as a resource to help one prepare further. You can buy a Membership card from them at the beginning of the year for a small fee, which will grant you access to all their non-seminar events. My recommendation, in terms of their seminars, is to try and attend at least a couple before deciding whether or not their review seminars are conducive to your studying.

8.2 CSEC

CSEC, the Computer Science Enrichment Club, is a relatively new club at UTSC, which serves to further your understanding of computer science, while teaching you concepts and skills you may not learn in your typical classroom experience.

CSEC-A, the Algorithms Division, goes over theoretical concepts and skills that will benefit those who attend Coding Competitions, but also serves to teach those who aren't familiar with said concepts as well. The CSEC-S division serves to

teach Network Security, and gets people familiar with CyberSecurity fundamentals. The CSEC-W teaches Web Development for both newcomers and those who might need brushing up on their knowledge, equipping people with skills that can benefit them in the workforce, and at Hackathons.

Each division meets once a week, every week, and hosts occasional seminars dealing with other topics like building a PC and perfecting a resume for Job Applications.

8.3 Hackathons

As mentioned before, Hackathons are a great way to expand your network and your skills. You may be asking, what exactly *is* a Hackathon? Simply put, Hackathons are competitions that usually occur over the weekend, either over the span of 24 or 36 hours, in which you and a small team work together to make something, be it an app, a webapp, or a hack using some piece of hardware, that you present and compete with at the end of the time allotment they give you. Hackathons are a great way to dip your toes into the world of competitive Computer Science, and as mentioned before, a great way to increase your network through the people you meet and the sponsor companies you encounter, but also acquiring lots and lots of free merchandise and t-shirts.

If you're willing to lose a bit of sleep and go to your first hackathon, ask around to see if anyone else is going to any upcoming Hackathons, or try and find one of the various Hackathon Facebook groups. Some notable Hackathons to apply for are Hack The North, HackPrinceton, MHacks, UofTHacks, and UTSC's very own Hack the Valley.

One more thing, while Hackathons are fun to go to every now and then, don't let them consume your life. Your education is what's most important (Aside from your health), and it shouldn't have to be marred by the amount of Hackathons you go to.

9 Conclusion

In the end, your University experience is yours, and yours alone. Whether you want to spend it holed up in your room studying, or making memes, no one can truly tell you how to live your life. Hopefully this handbook has been of some help to you, and the fruits of my boredom provided you with some new insight. Take a deep breath, and head forward to what awaits you, because it'll be one hell of a ride.

10 Useful Links

UTSC's Meme group:
www.facebook.com/groups/326488927723078/

UTSC Class of 2021 Facebook group:
https://www.facebook.com/groups/1870631783212775/

A repository containing textbooks you'll need for first year, and more:
drive.google.com/drive/folders/0BxrdNKPmoDq-N2s0ekxlY0NWd0k

Dates and Deadlines for the 2017/2018 school year. Useful for figuring out when things are due:
hive.utsc.utoronto.ca/public/registrar/Fall%2017-WInter%2018%20dates.pdf

The Program Page for Computer Science. This will tell you what courses you'll need in order to graduate:
utsc.calendar.utoronto.ca/specialist-program-computer-science-science

An official Timetable planner that you can use to plan courses:
https://ttb.utoronto.ca/ttb/#!/

11 Making this Handbook Better

As this handbook was made over the course of an afternoon, it's likely to be riddled with typing errors and formatting errors here and there. Finding those errors and fixing them will only make this handbook better and more robust for future generations. If you do find any errors, feel free to email kohilan6@hotmail.com with the subject line **CS Handbook Revision**, and we can work towards getting it fixed. A plus side of this is you can get your name added to the list of contributors to this handbook, and perhaps may even get the responsibility of updating this handed over to you one day. Until then, I can only thank you for taking the time to read through this all.