# (COEN) Advice for (Incoming) Students

# I. Introduction

Let's start with the title–"COEN" and "incoming" both represent target audience aspirations rather than hard and fast rules. While I'm certain that a number of the things that I'll be discussing in this paper will translate to other students in other years pursuing other majors at other schools, the goal here is primarily to produce the kind of paper that I wish I could've read as a Computer Science and Engineering freshman at Santa Clara University. Furthermore, this paper is **not** intended to cast judgment on those who hold different ideals, strengths, and motivations than me–it's simply a slice of history as to how I managed to do well in my niche, written out while the experience is still fresh in my mind (during the winter and spring quarters of my senior year).

I intend to cover a variety of topics in this paper, including navigating classes, work loads, on-campus clubs, honor societies, personal projects, research opportunities, grad schools, corporate internships, traversing SCU as an institution, capitalizing on the resources afforded to you by your tuition, and more. Additionally, any and all views expressed in this paper are exclusively my own as an individual. While I (of course) won't be pointlessly disparaging any person or class in this text, I also won't shy from giving genuine advice when it comes to navigating some of SCU's challenging components (blessedly few as they are).

Should any questions arise after reading this piece, please feel free to contact me at **jordanran199@gmail.com**.

---

# II. My Background

Before jumping into it, let's start with a bit of background information on myself, so as to paint a quick picture of the person behind the words that you hear in your head as you read this, as well as to establish what I've been able to achieve by following the advice below. My name is Jordan Candide Randleman. I joined SCU in the fall of 2018 as a COEN major from the start, and I'm slated to graduate in June 2022 with a minor in MATH as well. I knew that I wanted to excel academically at college from the very get-go, and despite coming in with limited free time due to having zero credits upon entry, I nevertheless made sure to make time to get involved with personal programming projects from the start of my freshmen year (a habit that I've maintained to this day).

In my early time at SCU, I accumulated a solid reputation among my peers and professors by extensively participating in class and collaboratively learning with/helping my classmates. In my junior year, I became an Engineering Peer Advisor (someone who helps new engineering students navigate SCU) and got inducted into both Tau Beta Pi (engineering honor society) and Alpha Sigma Nu (Jesuit honor society). Then, as a senior, I became both the Lead Engineering Peer Advisor and President of Tau Beta Pi (California Zeta), got inducted into Sigma Xi (scientific research honor society), got inducted into *and* became the President of Upsilon Pi Epsilon (computer science honor society), and was one of two recipients to be honored with the *Outstanding CSE Senior Award*, which is "*Awarded to a senior or seniors by the faculty of the Computer Science and Engineering Department based on academic standing, esprit de corps, and contribution to the department, school, and community*".

With respect to corporate experience, I interned at a then-startup in Redwood City named Wynd Technologies during the summers after my freshman and sophomore years, which I initially got in touch with via a family connection. I then interned at Microsoft during my junior year's summer, after first catching the eye of a career fair recruiter with my personal projects, then solidifying my potential with my prospective team by passing three technical interviews and going into depth regarding my time at Wynd. Despite receiving a full time offer from Microsoft's Azure division ($30,000 signing bonus + $145,680 TTC/year + stock options), I decided to turn the offer down in order to pursue a Master's at SCU, which I currently intend to follow up with a PhD in Program Synthesis from another university (I actually initially intended to get a PhD immediately out of undergrad, and was accepted at UCI, UCSD, and Georgia Tech, but for personal and professional reasons decided to stick with SCU for two more years before diving into a PhD once again).

All of this name and number-dropping may come off as somewhat distasteful, but this is simply to illustrate the kind of promise that a COEN future can hold at SCU. Note that, as far as I've seen, this is *generally* **not** a feeder school to top tech companies and top grad schools for *most* average students, but it does appear that *most* top performing students (with respect to academic and technical chops) have *relatively* little struggle in finding excellent opportunities post-graduation–so long as they are proactive in seeking such out. **Again though, like the rest of this paper, these observations are 100% anecdotal and I have no hard data to back them up.**

Each of these attainments was how I parameterized my own "success" throughout the years, in addition to striving for mastery over course materials. In my time talking to underclassmen, there've been many folks who want to do well in school, be a good

programmer, get a big-name internship with a return offer, and/or be accepted at a solid grad school. **While there's obviously no way to guarantee any of these milestones, there are absolutely steps that can be taken to increase the chances of your success in achieving them**. Hence the impetus for this paper–to act as a reflection from the perspective of someone who's treaded this path, with advice to give but only so much short-term memory at their disposal when trying to recall things off of the top of their head while going over tips with an ambitious freshman. Furthermore, if even just one student finds something of value in this piece that helps them succeed at attaining their goals, writing this entire paper will have been worth it :)

---

# III. Common College Misconceptions

### *i) "Work Hard, Play Hard"*

Being an engineering peer advisor has been quite enlightening–since every new engineering student automatically gets one assigned to them by SCU, peer advisors get to meet a lot of new folks and see how they've framed the concept of "college" in their mind. Some come in expecting to be able to live the "work hard, play hard" life as an engineer, typically as a result of being misled by Hollywood, the internet, or their parents (or family friends) that graduated in a non-engineering discipline. **This generally ultimately finishes in a horrible crash and burn scenario** where the student panics, realizing that they've left a permanent (and completely avoidable) scar on their academic record, thanks to screwing around while pursuing what many consider to be a notoriously complicated set of subjects. This is not to say that you can't have fun, rather that you can't regularly intoxicate yourself and skip class/not study while expecting to still come out of college even remotely competent in your area.

### *ii) "Work Smart, Not Hard!"*

Equally problematic to the "work hard play hard" mentality is the infamous "work smart, not hard" motto. Incredibly popular in Silicon Valley, **this is terrible advice** and an insane statement to make. Smart kids that don't work hard are functionally worthless to the world due to not fully capitalizing on their cognitive capabilities to manifest meaningful results, and banging your head against the wall without trying to optimize your approach will only serve to give you a

migraine. It sounds obvious, but seriously, **you will have to work both smart and hard to realize results worth writing home about.**

### *iii) "Fake it Till You Make it"*

Another dangerous quote common in Silicon Valley, this one is a bit more nuanced and depends **heavily** upon its interpretation. The charitable view is that it says you shouldn't feel insecure about yourself off the bat, rather just keep playing the role you're in until you feel like you fit in after acquiring more experience. However, this (to me) takes away from the important data that your insecurity provides–**don't just ignore your feelings of inadequacy, instead analyze them to determine what you should focus on in order to get the skills that would make you feel more secure.** By doing so, you won't be "faking it" anymore, rather you'll be in the honest and self-aware process of self-improvement in order to be at the standard to which you perceive those around you to have already achieved. I've also heard one person at Microsoft put it this way when it comes to dealing with fears about your place in an organization: "if you don't have faith in yourself, at least have faith in those who put you here after evaluating your work and personally vouching for your skills".

**The sinister aspect of this saying comes in when you take the "fake it" part literally**–in my opinion, you should **never** say you have something done or fully understood before such is actually the case. Proponents will point to Bill Gates' famous OS situation, but I have two points to refute such: (1) you're not Bill Gates (2) Look up "Elizabeth Holmes Theranos". If you're supremely confident in your ability to get something done quickly, just say so! But don't lie to those depending on you by concealing the truth of your inability and/or inaction, as it only serves to destroy your team's ability to coordinate effectively in good faith.

### *iv) "I'm Just Here for the Piece of Paper"*

Sometimes students come into the engineering school with the idea that they are "just here for the piece of paper" to graduate–and indeed, if that's all you seek from the institution, that's all that you'll be getting (if even that, as a general disdain towards college tends to cause apathy towards courses–and generally low performance–that is unconducive towards graduation). The proactive application of the resources at your disposal is absolutely critical if you are going to crush college and actually learn something, for expecting to walk in and out of class while being spoon-fed material to "technically pass" will rather fittingly leave you little better than a child.

### *v) "I Have to Take How Much STEM?!"*

And still others come in perplexed by the notion that they'll be taking 4-5 classes per quarter, only one of which isn't STEM. This one genuinely surprises me–when I entered into Computer Science and Engineering, I expected it to be hard as nails! Have you seen a computer? It's a complicated machine! And yet, likely due to the generally dismal state of the American High School education system (I have no clue how it is abroad, and to be clear I was very fortunate to attend an excellent high school [Waldorf School of the Peninsula]), a number of students come into college facing the first real academic challenges that they've had in their entire lives. College is a period of time in which to invest in yourself intellectually: you possibly have the most physical stamina, mental flexibility, and least responsibilities that you'll ever have for the rest of your life–make sure to capitalize on such by learning as much as you can in order to set yourself up for success throughout the rest of your intellectual and corporate careers! That's going to mean taking a number of hard classes, so as to accrue experience in a variety of STEM domains, something that you can't do in the span of four years without at least **some** degree of intensity.

### *vi) "Who Needs Notes?"*

**You need notes.** While most (emphasis on *most*) freshman professors will take pity on you and provide study guides or upload their lectures, come your later years at SCU and you will be expected to show up to class on midterm day ready to fully demonstrate your understanding of all of the material you've covered thus far–no study guides, no lecture recordings, no pre-provided notes, none of that. Now, maybe you're the exception and happen to have perfect recall–except no, you absolutely don't. Therefore, taking down meaningful, hand-written notes (to increase retention) with color-codes to highlight important sections (for easier skimming upon later review) will be critical. **Your notes and old tests from class will serve as your study guides for later exams.** Furthermore, you'll also want to annotate your notes with the current date **and week number** in order to better line such up with course materials posted on Camino. And finally, you'll also want to develop shorthands to take efficient notes, as you don't want to be so preoccupied with writing everything down that you can't properly participate in the classroom.

### *vii) "I'll Just do the Homework at my Tutoring/Study Session"*

**Not if the tutor is doing their job you won't.** In order to do well in school (and life in general), you will have to develop the ability to wrangle with hard problems to synthesize solutions on your own, in conjunction with the application of in-person and online resources. Go to a tutoring session in order to learn fundamental concepts that you'll be able to apply when finishing your homework later on your own time, and use student study groups to compare approaches that different people took to common problems in order to learn from each other's techniques.

Elaborating on study groups, I had one peer that used to clearly only show up in order to get "the smart kids" to do their work for them, and low and behold they had to drop the major once the rest of us got tired of supporting their inattentiveness and cut them off. Note that this served to save everyone's time, including the student who dropped–if someone is struggling to pay basic attention in intro classes, coupled with a general sense of apathy towards the course material, they ought to strongly consider leaving sooner than later in order to find another domain that better suits their strengths.

**Furthermore, know that going into office hours or a tutoring session right before an exam and expecting to learn everything from 5 weeks in 2 hours is academic suicide**–there is a reason it took the professor 5 weeks to convey the material! At three one-hour classes per week, the student tutor would have to have an 86.67% compression rate for the course material, *which isn't happening.*

### *iix) "Nobody Cares About Your Grades in College"*

This one is especially popular with disingenuous internet personalities that need a clickbait title, as well as with recruiters that are overconfident in their estimation regarding their corporation's ability to catch students up with what they already ought to know heading into a position. While some tech companies genuinely couldn't care less about your ability to do well in Art History and instead only care about concrete technical skills, the reality is that it is exceedingly rare (in my experience) to see a poor-performing student at SCU who systematically focuses in some areas and slacks off in others so as to maximize their cultivation of hard-core technical skills in their spare time. While I want to be careful not to use too broad a brush, in my anecdotal experience, top students tend to be such because of their intense drive and determination to achieve excellence (which generally translates to developing technical skills outside of class as well), whereas students that do poorly tend to be too wrapped up in not caring and/or being hyper social (which rarely incorporates any extracurricular engineering whatsoever). Again, this is not a clean and cut rule, and I can think of multiple students who **are**

incredibly technically competent while still struggling academically–however, they are far and away the exception rather than the rule.

Furthermore, the technical teams that'll give you interviews at top tech companies are generally run by smart folks who almost certainly also tried hard in school, and as such, they may have an implicit bias towards those candidates that are demonstrably invested in being the best, high-achieving student that they can be. Additionally, while there are companies that don't care as much about academic performance, you better believe that grad schools care, and that you'll almost certainly need some seriously impressive research in order to overcome the stigma of a 3.0 SCU GPA (note that a 3.0 is the bare minimum GPA expected for most grad-school applicants).

### *ix) "Everybody Cares About Your Grades in College"*

On the flip-side of the previous misconception, we have the fear of not attaining a perfect 4.0 GPA. While striving for academic excellence is highly commendable, college can only take you so far. The way to frame college (from a COEN education perspective) in your mind is as a means by which to get introduced to a number concepts that you hadn't even heard about before, to learn the basics of said concepts, and to get access to those who've been working with these concepts for a number of years (which includes professors, peers, and upperclassmen).

However, **you must also take the leap to go beyond the scope of the classroom to explore each of these subjects further on your own**. College allows you to dip your toes in pools of knowledge, and your GPA will only reflect how well you did at the said dipping of toes. I would not be a third of the programmer that I am today if I didn't use challenging personal projects to expand my understanding of classroom and internet concepts. It is infinitely better (in both my estimation and that of many others) to come out of college with average grades and an intense appreciation for the pragmatic and theoretical applications of computing, than to have a perfect scholastic record stained by practical inability. Furthermore, while this is almost certainly obvious to anyone who'd take the time to read such a piece as this, I nevertheless still want to make it absolutely clear that **high grades and solid technical skills (from extracurricular activities) are by no means mutually exclusive**, and that the combination of both makes for an intensely powerful (and highly desirable) set of skills and accomplishments.

### *x) "I'll Do The Work Later"*

Be extremely wary of unnecessary delays in the accomplishment of tasks. Don't get me wrong, pacing yourself is a valuable skill, and one I frankly ought to have worked on developing earlier (I lived by a "do everything immediately" motto to an extreme degree as a first and second year student). However, there are multiple risks at play when you try to kill all your tasks at the last minute:

a) **Workload time frames are not static.** A professor may decide to give you another assignment or pop quiz to study for at the last minute, causing your time management assumptions to fall to pieces if you were relying on that weekend to complete an outstanding task.
b) **You're assuming that you won't need any help.** Tutoring and office hours aren't 24/7 services, and hence you'll forsake the ability to capitalize on these resources if you start your work too late to meet up with them prior to the assignment deadline.
c) **You're putting yourself through unnecessary stress.** Trying to do everything at the last minute is like trying to finish a marathon by only sprinting–jogging can make the process practically pleasant, and without the risk of a heart attack.
d) **You risk becoming susceptible to dangerous habits.** Folks who do things last minute are more likely to become dependent on online solutions and peers for answers. Both of these resources can be great to verify your work and understand what went wrong in order to learn more about the material, but blindly copying them will do naught but screw you come exam time.

## *xi) The Cancer of Class Competition*

The last misconception that I'll be addressing is the idea that your class peers are academic competition. While in truth the majority of classes at SCU do use a curve, such is completely irrelevant if you are striving for a perfect 100% score in the class, ***which you should always be doing anyways.*** Furthermore, ignoring the curve in order to perform well on an absolute basis (as opposed to a relative basis) opens up the door to unrestricted cooperation with peers, meaning that you'll each get to learn from each other and improve your skills without fear of such backfiring on you. Note that SCU is actually fantastic in this regard: its small community genuinely does foster a cooperative environment that favors friendship and collaboration over cut-throat competition with peers–after nearly four full years here, I can confirm it's not just marketing BS.

**With that being said, there are scenarios in which you do want to play it "close to the vest", so to speak.** Whereas everyone could theoretically get 100% in a class and earn an

A, the reality is that there are only so many positions available at corporate internships, grad school PhD programs, and research paper journal publications. Academic knowledge is an endless fountain that can be shared with everyone, but the opportunity to apply such at any given institution with limited capacity is not. Be helpful where it makes sense and support those you cherish most, but don't be naive and shoot yourself in the foot by advertising a scarce opportunity to those who don't bother seeking such out for themselves.

---

# IV. Academics

## *i) Physical and Mental Health*

While it might be strange to see this as the first aspect of academic success discussed, without either of the above, you will fail sooner than later and potentially cause immense, completely avoidable, and nearly irreparable harm to yourself. **You should approach college academics like an athlete prepping for a game.** This includes (in no particular order):

a) **Exercising Regularly:** I'm not saying you need to do 500 pushups or run a half marathon every day, but you should absolutely be exerting yourself physically at least to some extent at regular intervals throughout the week. Engineering is largely done indoors, sitting down (though I encourage the use of a standing desk, I used to put my chair facing me on my dorm desk to create a sort of pseudo-podium), under cognitively intensive/stressful circumstances. Exercising gets you outside, moving around, and helps relieve tension. The hardest part is starting, but cultivate a multi-day per week habit and you'll thank yourself down the road (just be safe about it!).

b) **Get Sleep:** It should be obvious by now to society that so-called "all nighters" are a ridiculously ineffective way to study, but the slow deprivation of sleep by cutting down on a few hours every night also begins to wear on you throughout the quarter. You'll become more snappy/aggressive and less attentive, so make sure to get your 8 hours. I've found it infinitely better to go to bed earlier, and study early in the morning with coffee, than to struggle staying awake till 3am operating at 50% capacity.

c) **Eat Healthy and Drink Water:** Not to sound like your parents, but seriously, most people are dehydrated like crazy–start intentionally drinking cups of water (not only caffeine) in the morning and afternoon, and in conjunction with a healthy diet, you'll start feeling significantly better throughout the day.

d) **Find a Support Network:** This can be friends, family, support systems at SCU, an online community, nearly anything. It's hard to meet people at first in college (feeling crazy lonely in your first **few quarters** is way more common than you might think), but it is critical to put yourself in "social circulation" by intentionally exposing yourself to new acquaintances by attending events hosted by your CFs (SCU's version of RAs), engineering peer advisors, SCU's faculty, and student-run clubs! Make an effort to study in areas that are popular for your major (this includes Heafey's Atrium and SCDI for engineers) in order to run into folks from your classes, thereby meeting the other friends that they hang out with, etc. until you create your own friend group. Note that you'll also passively accrue contacts by attending classes and study groups over the years, albeit in a **significantly** slower fashion than the alternatives I mentioned above.

e) **Understand Why You're Doing This:** Being an engineering student can be extremely intensive–you have to have a rock-solid understanding of your own personal motivations for putting yourself through so much stress. Without a clear appreciation of the factors driving your decision, you'll swiftly fall to apathy and plummet in terms of performance. Personal motivations can be as altruistic as wanting to create technologies that help the world, to being entirely selfish in desiring a fat paycheck. Maybe you just joined because you heard it was a popular field, or because your relatives insisted that you do so. I would argue that none of the above are sufficient reasons in and of themselves–you need to be fundamentally interested in learning to integrate science and mathematics as tools in the creative process (with a problem-solving bent) in order to truly succeed and thrive as an engineer. You won't be able to fully help the world if you're miserable, and you can't take cash with you once you're in the dirt (or fired out of a cannon, as I intend to go). Part of college is learning to become an active, independent agent that can think and operate for themselves, and while taking advice is great, letting those around you live vicariously through your decisions is not. **If you aren't personally interested in solving problems to have fun while using STEM as a tool to launch humanity into a better future, I put forth that engineering is almost certainly not for you.**

### *ii) Attending Lecture and Office Hours*

**Never skip class unless you're sick or dead.** Arrive early, review yesterday's notes prior to today's lecture to refresh yourself on background material, sit at the very front of the class, and ask questions like mad. SCU demands a quarter-million dollars for its 4-year tuition, and yet, a number of students (especially in freshman/sophomore classes) seem to be giving as

much attention to the professor as they would a YouTube video running on a background tab. Under no exception (bar a developing emergency) should you be screwing around on your phone or laptop during a class period.

**Make sure to attend office hours and tutoring early**–everyone under the sun will line up the day before the midterm, and all that does is tell the professor “I’m one of the students who is barely paying attention and can’t be bothered to keep up with your lecture in real time”. You don’t want to plant that seed in the professor’s head, and you don’t want to have to cram material at the last minute (material you’ll likely barely be able to cover given the number of other late students you’ll be competing with during the meeting). Be proactive in your intellectual journey as you navigate your courses by understanding concepts and applications as they arise rather than at the last minute.

**You should refuse to ever leave a classroom ignorant to what you were just taught**–keep asking the professor (it's their job) and thinking about what’s going on until you fully understand the material. Once, in Physics 33 (electricity and magnetism), the lecture professor was walking the class through some complicated math. However, having had issues following along with the lecturer’s reasoning, I went with a friend to their office hours in order to walk the professor through our confusion, only to confirm that the lecturer had in fact made a mistake in their application of mathematics to the physical system in question. Out of the 40-odd students attending lecture, only two of us were actually willing to either pay enough attention to the course to notice the mistake, or even more insidious, to speak up about such in order to help our peers by triggering an announcement about it the next day.

Moreover, **there are many students embarrassed to ask supposedly “dumb questions”.** This, to me, is as deeply unfortunate as it is silly, for the following reasons:

a) Everybody will have forgotten you asked the question in 60 seconds tops, so even if it is embarrassing, **nobody cares in the long run.**
b) By refusing to ask a 30 second question in class, you may have to spend 30 minutes trying to figure out an otherwise trivially explained concept back at your dorm for homework.
c) In all likelihood, there are a number of people who also have the same question as you, so by not asking, you’re not only doing yourself but the entire class a disfavor.
d) As ridiculous as you worry you might sound, how ridiculous is it to do poorly on a test question that you knew you didn’t understand but were too embarrassed to ask about, simply because of some abstract notion of an ephemeral judgment by a peer too ignorant to appreciate the value of capitalizing on your tuition?

### *iii) Attending Lab*

**You should be trying to do as much of the lab work on your own ahead of time**, especially for programming-centric courses. You can't just punt your problems to a TA in the corporate world–you'll be solving problems that generally aren't solved yet, and hence will be expected to be able to put systems together on your own (as done every week by the folks who try and finish their lab assignments ahead of the allocated class period). Furthermore, you'll get to be one of the cool kids who just shows up, submits, and either hangs out or dips as they please 8^)

**Be extremely careful when going through your code with your colleagues in order to explain fundamental concepts.** If you're the kind of student that is consulted for help by others, you're almost certainly a smart cookie. As much as you might like to think otherwise, you cannot assume that everybody else is as well. **I have witnessed multiple instances of smart students getting screwed over because the person they were trying to help just straight up ripped their work off** without making any meaningful changes. This in turn got the helping student academically reprimanded for facilitating dishonesty, despite the fact that the helper only had the best of intentions in mind.

**Should you go over code with another peer** (which I strongly discourage but undeniably happens), make sure you **significantly change your work beforehand** so that you will not be caught in the crosshairs of the institution for facilitating a potential cheat that you were completely unaware of as a bad actor. SCU has an incredible number of kind/honest students, but **every institution of meaningful size will have its share of disingenuous individuals**, who will lie and cheat and steal to get ahead. **It is your responsibility to protect yourself against these bad actors, as well as to report them to the professor as soon as you realize what they're doing.**

**Should you be caught up in a situation akin to the aforementioned, immediately contact the professor and be completely forthright regarding what happened**–you don't want the cheat to be able to plant the narrative seed in the professor's head before you, as if they're willing to rip off your work, who's to say they won't throw you under the bus too?

### *iv) Interdisciplinary Computer Programming*

There comes a point in every programmer's life (or at least every programmer dedicated to their craft) where they begin to appreciate that programming can be used as a generic problem-solving tool–it shouldn't just be relegated to a classroom's problem sets!

I was fortunate enough such that this realization dawned on me in my freshman year–throughout my time at SCU, I always strived to incorporate programming in my understanding of (and exploration into) non-CS subjects as well. In MATH 13 I would program vector mathematics in order to explore the results of various calculations, and in ELEN 21 **I made a webpage** dedicated to generating Truth Tables, KMAPS, and SOP/POS forms of equations to verify my homework. I partially credit my success in PHYS 33 to my programmatic simulations of electrical charges so as to verify that I fully understood what each variable in each equation meant (note that these weren't hardcore GUI simulations, rather just performing algorithms to measure results on points represented via structs with "int" fields for the X, Y, and Z coordinates in space). Furthermore, while not literally encoded, I found that object-oriented programming made for a fantastic mental model to encapsulate the various entities that physics has to offer, a demonstration of just how powerful programming can be–even without a computer on hand.

**Unlike most humans, a computer will never accept a "hand-wavy" solution to a problem–you have to be able to explicitly line out exactly what every datum is, and what it does in relation to other data and itself, in order for your code to run.** As such, making sure that you're able to programmatically codify a concept in a machine makes for an excellent exercise in validating your confidence towards a particular subject.

### *v) Registering for Courses*

Course registration can be a notoriously stressful time for students in general, let alone for those in an impacted major like SCU's COEN degree. However, let me start off with a caveat: for my first three years at the university, SCU gave course registration priority to those who had more units to their name. As a result, having come from a highschool with a graduating class of 10 people and no AP classes to speak of, my registration times were absolute trash every single year–meaning that I had to get creative in order to secure the courses that I needed. **However, SCU has now eliminated this practice, and as such everyone ought to be on a more even playing field.** With that being said, I did still learn a number of things in my time, which I've outlined below should folks still need help:

a) **Try to follow the four-year plan.** You don't have to follow it religiously, but the closer you stick to it, the easier it will be to avoid confusion regarding class prerequisites. **Not every class is offered every quarter at SCU**–deviating from the four-year plan means that you'll have to manually account for such when restructuring your schedule.

b) **Actually use the advice from your school-assigned engineering peer and faculty advisors.** Every engineering student is required to attend fall and spring advising sessions hosted by the engineering peer advisors–these sessions are intended to provide you with a list of SCU's various academic and social resources on campus, as well as to help you outline your schedule over the next few quarters. *This isn't done just to waste your time, it's designed to avoid having you pay an extra $60K for a 5th year here after making a completely avoidable logistical goof.*

c) **You will not immediately get into all of your intended classes.** I spent the majority of my quarters at SCU regularly missing out on two to three of the classes that I needed (despite being perfectly prompt for my registration appointment), but *in the end, I always managed to secure them* with the help of professors and my faculty advisor. Just breathe, don't let it get to your head, and follow the advice outlined in this paper.

d) **Be extremely prompt to your registration appointment.** Once you're at SCU you'll find out how to find your class registration date and time (referred to as an "appointment") on eCampus, which denotes when eCampus will officially let you add a new class to your schedule. Note that this is **not** the time at which you should start adding classes to your cart (you should have already done so ahead of time), rather it's the time 10 seconds before which you should start mashing the "submit" button to make sure you get all of your courses added ASAP (this is absolutely imperative, as classes with 20+ seats can disappear in the wink of an eye if you lag).

e) **Make multiple schedules on courseavail, and recheck multiple times** over the coming days (and hours) before your sign up time to make sure that your plans are still viable. You'll almost certainly find yourself ripping apart and stitching together schedules to account for class changes as seats dwindle over time.

f) **Check the box on eCampus to be put on a class's waiting list if it's full** when adding it to your cart for your registration appointment. This is step zero to getting into a class that's currently full, and without doing so, you'll just look like you're wasting the professor's time when you email them later for a spot.

g) **If you don't get into a required class, *immediately* contact your assigned faculty and peer advisors, as well as each professor for each class that you didn't get into.** Don't just sit on your rear and do nothing–that's how you accidentally spend five years here, or worse, get kicked out for being a part time student for not having enough units to your name in a quarter. When emailing your professors and advisors, make sure

to clearly articulate precisely why you need that class at that time, and include pictures of your current schedules to prove its necessity.

   i) **Note that this is not to be done with classes you just want to take for fun** (you'll only irritate your contacts for wasting their time with the non-essential), rather it's meant for classes that you need to graduate on time. **And for the record, *"I don't want to take an 8am"* is not a valid reason to ask for another spot in a different class that is already full.**

h) **Show up on the first day of class early to catch the professor before and after their lecture to demonstrate your dedication to the course you didn't get into.** Note that this practice is technically frowned upon by the engineering school as an institution, but it is smiled upon by students for whom it has worked, and I'm writing this paper to give advice as a peer rather than as an official advisor.

i) **Advanced students will have an extremely hard time getting into advanced courses.** Seniority still grants priority in class registration (to make sure everyone graduates on time), though extreme persistence with a professor can *sometimes* help (don't be too irritating though, as you don't want to squeeze into a class at the expense of your reputation with its professor). For reference, I do know one student that managed to eventually get into an Operating Systems lecture as a sophomore despite initially being rejected by its lecturer three times.

j) **Use the "ratemyprofessor" website and official SCU course evaluations in order to get a small insight into the professor's teaching style before adding a course.** Note that both of these resources should be taken with a healthy dose of salt though, as sometimes students get enamored or enraged with a class for the wrong reasons.

k) **Consult upperclassmen for advice when choosing a course.** If you don't know any upperclassmen, you can always contact your assigned peer advisor, as well as show up to an ACM meeting (or any other technical club) and randomly ask the folks there for tips too. You might gain valuable insights as to what/who you should and should not take, with a particular ELEN professor (who shall remain unnamed in this paper for the sake of politeness) being especially notorious–ask any upperclassmen COEN, and they'll know exactly who I'm referring to.

l) **Courseavail's seat estimates are not perfect.** While rare, sometimes it will say that a class is open when in fact it's full, and ultimately only eCampus truly reflects whether there are any slots left in a class. Furthermore, if you ever wonder why some classes show "-5 seats left" on courseavail, that's because class sizes are artificially constrained

on paper in order to save room for students who really need a spot at the last minute to graduate on time, and hence the negative number shows that students got in through these reserved seats beyond the advertised capacity of the course.

m) **Beware of "midterm stacking".** This occurs when you try taking a bunch of hardcore classes in one day, thinking you can handle a lot of consecutive lectures, without appreciating that such means you'll also risk having all of the midterms for those classes on the same day as well.

n) **Double-dip CORE requirements as much as possible!** SCU engineers have the unique ability to satisfy multiple CORE requirements in one CORE class, allowing them to significantly ease their class burden over the years (note that CORE classes are general education classes that you'll have to take to graduate in addition to your major-specific courses). It's not the end of the world if you take a non-double dipping CORE class, but **every double-dip class you take is one less class you have to take later on!** You can get the list of double-dipping classes by asking your peer advisor for it, and you can verify it's still accurate by checking course descriptions on courseavail after searching for a class and clicking its course number for a pop-up with further information.

### *vi) Intentionally Choosing Hard Classes*

There may be a temptation to take easy classes that boost your GPA, at the expense of taking harder classes that could compromise such but would nevertheless teach you more. I've heard of such classes jokingly referred to as "underwater basket-weaving courses" (not to disparage divers or basketmakers mind you) due to their focus being on the development of a questionably applicable/niche skill set. Do not take these classes. **Shying away from meaningful courses is a sign of intellectual laziness and insecurity that is to be firmly condemned and shunned.** A GPA is only as valuable as the classes taken to earn it, do not pollute your record with bogus courses to give a facade of intelligence and/or competency–**hardcore tech companies like Google will ask for your transcript to see what classes you've taken**, and I promise you that they'll see right through any disingenuous attempts to articifially bolster otherwise dismal academic performance.

Furthermore, if you're worried about throwing yourself in the deep end of a subject that you don't have an intuition for, **you can cultivate an intuition for hard topics by repeatedly exposing yourself to practice problems outside of class in order to ingrain a familiarity with the subject's problem solving process.** For example, whereas math had always come naturally to me, I had to do this for my physics courses by going over a number of practice

problems every evening in order to instill a sense of internalized familiarity with the approaches employed by the physics examples.

### *vii) Putting Effort into CORE Classes*

There is also the question of whether or not it is worth trying to get a good grade in a class completely disjoint (at least at first glance) from your major and future ambitions. While I know folks that I consider to be extremely clever on both sides of this debate, I would argue that trying hard is absolutely worth it, for the following reasons:

a) **You'll develop other skill sets.** STEM classes generally won't exercise your capacity to express yourself in an especially coherent way to other human beings, rather focussing on quantitative results-oriented metrics (this is in fact part of the beauty of STEM). However, an astoundingly high number of my peers' writing reads rather questionably–having had to perform several essay peer reviews, I find myself wondering why they sound so smart when they speak but write as if they have a concussion. Developing the ability to operate in novel circumstances, with completely different groups of people, will serve you well as you navigate the ocean of life–that means exposing yourself (and trying hard in) non-major specific classes in order to cultivate such.
b) **Grad schools, honor societies, and most accolades will use your cumulative GPA as their metric for academic success**–something affected by CORE classes.
c) **You'll sleep better at night.** Note that this is of course *highly subjective*, but personally, I can't stand the idea that my performance looks worse than it otherwise could have because I didn't try harder at a subject that I know I could've trounced with relative ease.

### *iix) Changing (and Adding) Majors (and Minors)*

Every year, there are a number of students that change majors within the engineering school, go to the business school/college of arts and sciences at SCU, or explore other universities. Their reasons are as myriad as the fish in the ocean, however, one ought to exercise extreme caution when doing so–with **the greatest sin being to transfer without consulting your peer and faculty advisors first.**

The primary concern regarding transfers is that they are doing so for the wrong reasons. For example, the physics trifecta of 31-33 is notoriously hard for COEN students, and for many of them it'll be their first C or D on a test. However, **transferring because of your grade in a class on the periphery of COEN is (to my eye) deeply unfortunate and fundamentally misguided**–physics is one of the hardest class sequences that you'll have to take at SCU, so

don't quit near the top of the difficulty curve! It's one thing to transfer because you struggle with the easy/average courses, but be extremely wary of tapping out when everyone else is having a hard time too!

**Another example of improper transfer motivations lies in that of a student who switched universities (but kept the same major) because they had a terrible GPA at SCU**, and wanted to find an institution that had an easier COEN degree. The problem isn't that SCU's program is too hard, the problem is that COEN is an incredibly complex field that demands attention and effort, without which you will fail anywhere with a degree worth its salt. While I can appreciate the stress incurred by poor grades, the answer is to either buckle down on your major or change majors: not to try and find the nth random podunk college that hands out degrees to anyone with a pulse.

When it comes to internal transfers within SCU's engineering school, students can currently do so in spring. However, please note that **while people used to get into colleges with an "easy major" and then transfer into CS to avoid the initially competitive application process, this is actually a terrible decision nowadays** for SCU students, as it can be extremely difficult (and fairly risky) to try and get into the COEN program as an internal transfer. The best way to succeed at such is to have excellent grades in your first year, and to start preemptively taking COEN courses in order to show your competence and dedication–but even then, **your success is by no means guaranteed (even if you check all the boxes)**. As with most universities today, COEN being impacted (i.e. completely full) at SCU means that even fantastic students can be turned down for lack of space.

**When it comes to adding a major, such can be incredibly difficult to do as a COEN without taking extra classes over the summer.** The main reason for this pertains to the nature of "double-dips": whereas engineers can use one CORE class to accomplish two or more CORE requirements (something fundamental to easing your schedule), none of the other schools can do so. This means that if you want to get a double major in MATH, you'll have to take a foreign language, forsake your double-dips, and buckle down on a number of additional humanities courses just to get an extra degree in mathematics. While I do know one student who **did** pull a COEN/MATH double major off, they (a) were a beast to begin with, and (b) had to take the aforementioned summer-course route to fit everything in.

**When it comes to adding a minor, the most common one to add for COEN students is Mathematics**–should you optimize your schedule correctly, you'll only have to take **one** additional course in order to secure such! The MATH minor requires COEN students to take two additional upper-division MATH classes (not AMTH mind you, which is "applied

mathematics" as opposed to "pure mathematics"), in conjunction with an upper-division CSCI class. Upper division classes can be identified by having a course number in the 100's range, e.g. COEN 79 is a lower-division class while COEN 175 is an upper. COEN 179 (Algorithms) is cross-listed as an upper-div CSCI class, and you can swap out any upper-div MATH class for AMTH 106 in your schedule and still get credit for AMTH 106. This means that so long as you make time to take one more upper-div MATH class in your free time, **and ask your faculty advisor for the link to apply for the minor with the college or arts and sciences**, you'll secure the MATH minor with only one extra class!

Personally, I got my MATH minor by taking Cryptography (MATH 178) instead of AMTH 106, and Graph Theory (MATH 177) in my free time–note that both these classes are deeply tied to CS in nature, and are excellent courses worth taking irrespective of attaining the minor as well. Other minors are a bit more involved to achieve, but you'll get all the advice you need to learn about such by asking your assigned faculty and peer advisors for help.

### *ix) Collaborating with Professors and Peers*

**Knowledge acquisition is a team sport–while you can play it individually, you'll be at a serious competitive disadvantage by doing so.** Go to your professor's office hours even if you don't have specific questions about the course material–especially in weeks between midterms where attendance will be low, it's a great opportunity to get to know your professor more and learn about their research, as well as to understand their intellectual experiences and motivations overall. These are folks who've generally been in the field for quite a while–getting a STEM PhD or Master's is no joke, and you should capitalize on these experienced individuals' observations (especially given that such access is a nontrivial component of what your SCU tuition is intended to afford).

Student study groups (as previously mentioned) are also pivotal to knowledge acquisition, as you'll be able to compile a set of novel approaches to traditional problem sets from your friends in order to gain a better appreciation for the utility and application of each technique. You can form study groups with friends in class, however, make sure to also invite students who seem attentive and dedicated to their education by asking a lot of questions in lecture and lab, as they're likely to have quite a bit to contribute. Furthermore, you should **not** expect the study group to be your only time to study, as they can sometimes devolve into social situations (which is definitely fun, but academically unproductive). Becoming a capable engineer means learning to solve problems on your own (at least as an initial attempt before further

collaborative efforts), and hence you should only use the study group to analyze and reinforce the studying you did on your own before and after meeting up.

### *x) Navigating Intense Workloads*

**Your top two enemies as an engineering student are chaos and inertia**: disorganization and laziness will bury you alive in last-minute assignments and missed due dates. Use a calendar (Google Calendar works well) and a planner/to-do list (I have a physical planner and use a Google Docs to-do list). Check them religiously, and set alarms/reminders on your phone to automate the process of remembering what you need to do next. At the start of each quarter of SCU, I go through my planner after checking course syllabi on Camino and pre-populate any test dates, study times, paper deadlines, and homework checking prompts that I can (it makes for a boring 15 minutes, but pays off for 10 weeks–a classic example of delayed gratification).

**Do tasks sooner than later.** Unless you are strategically delaying a task, whether to coordinate with someone who's busy or to wait for incoming data, you should work on it as soon as you can. I always strove to be about a week ahead of all of my assignments in my time at SCU (with varying degrees of success, of course). There were multiple times where I've emailed professors asking them to post material early because I wanted to get it done ASAP, with the vast majority of responses being positive and impressed by my determination to do well.

As mentioned earlier, **do not fall into the trap of depending on online solutions or peers by blindly copying results**. While these resources can be used to verify your work and understand concepts, ignorantly regurgitating what you've found will only serve to cause you to perform abysmally on the exams that make up the vast majority of your course grade.

**Prefer working in the morning with caffeine** (though of course take care to consider how much you're consuming). You'll be at your mental peak around 10am, so don't spend that time goofing off and instead capitalize on your natural attentiveness. Additionally, you'll find it significantly easier to enjoy yourself without having the "oh crap I still need to do this thing" thought buzzing around in the back of your mind, so just handle your work sooner than later so that you can relax guilt-free.

The last tip that I have regarding workloads is to **eat your metaphorical vegetables first**. While task timelines will fundamentally determine what you do when, I've always striven to do work for subjects that I didn't enjoy as much first, that way I could relax by doing the homework for classes that I found interesting later on.

## *xi) Getting Support*

Make sure to seek (and accept) genuine advice from folks who've been designated to help you by SCU's engineering school as an institution–namely your faculty and peer advisors. While seeking advice from other peers and upperclassmen can sometimes bear fruit, institutional advisors receive *extensive training* pertaining to the optimization and nitty-gritty details of engineering schedules that the average student *does not have access to*.

**Before making a big decision** (say changing majors or dropping a course), please, please, **please make sure to contact your designated advisors beforehand**, as you may inadvertently quit for the wrong reasons or drop a set of classes that forces you to graduate in 5+ years. I'm being dead serious, you'd be amazed at how many new students seem to think they can completely forsake those who've done this before by falsely assuming that whatever they do, it will still work out in the end (from first hand experience, I can tell you that it won't without significant and wholly unnecessary heartache). The danger of dropping courses is found in the so-called "**prerequisite domino effect**"–not every class is offered every quarter at SCU, and some classes are required as prerequisites for others: it doesn't take a 300 IQ to realize the risk of dropping a prerequisite class that's only offered once a year just because you were scared of getting a B+.

In addition to advisors, being on the board of Tau Beta Pi allows me to see that the number of folks capitalizing on our chapter's free engineering tutoring services (which is advertised every week via emails to all new engineering students from their engineering peer advisors) is absolutely dismal. **There is a ridiculous stigma in society, and with STEM folks in particular, towards seeking help from others**–if you do not understand something about the course that you are taking (i.e. have less than 100% in the class) you have good reason to strongly consider attending a tutoring session hosted by either Tau Beta Pi (for engineering classes), the Math Learning Center (MLC for short), Drahmann Advising (for science and CORE classes), or The Hub (for essay and writing reviews) in order to clarify your confusion.

Furthermore, should the extremely rare case occur where you try attending a Tau Beta Pi tutoring session that the designated tutor missed, **please report the date/time and tutor you were expecting to taubetapi@scu.edu immediately**, as such is absolutely unacceptable and will be addressed by the board.

If you are one of the proactive students capitalizing on these opportunities, first of all, my serious props to you :) Second, there are a few things to keep in mind:

a) **Reach out sooner than later.**

   i) Tutors have busy lives too–don't expect to email them for help in the morning regarding a concept pertinent to an assignment due later in the evening.

b) **Tutoring is not available from Tau Beta Pi during week 1 or finals week.**
   i) We have to study for our own finals too, meaning that you'll want to go over course materials with us no later than by week 10. Week 1 is reserved to flesh out the delegation of classes to the chapter's tutors.

c) **Tutoring is not a time to do your homework or catch up on missed lectures.**
   i) It is entirely within a tutor's rights to turn down students who are just trying to abuse academic support services to get out of doing their own work.

d) **Don't be afraid to reach out to tutors via email for another time slot!**
   i) While not required of them, many tutors will take off-the-clock appointments to help, so don't be afraid to politely request a meeting with them if their allocated time slot doesn't work with your schedule!

### *xii) Giving Support*

Clever students with at least some social skills (not always a guarantee among engineers mind you [I say that in the most loving way possible]) will almost certainly be called upon by their peers for help with classes at some point. With this request comes a few risks and responsibilities that every student helping others must keep in mind:

a) **Don't enable cheating.** Putting obvious moral hazards aside, it leaves you and the student at **risk of being caught** and getting in serious academic trouble (something that doesn't even account for the reputational damage you'll incur with the class's professor). Note that some professors even use software to compare submitted code–this isn't just the job of TAs, authenticity can be verified programmatically. Furthermore, **cheating doesn't actually help the student**, as they aren't learning the concepts they need for both future exams and later on throughout their life. Lastly, **the student will come back to you again in the future to cheat even more–*it is never a one-time thing!***

b) Related to the above, **you need to learn to say no to people**. Disingenuous students trying to prey upon your good will shall typically use extreme confidence, extensive insisting, and/or pity-me guilt-tripping to make you feel social pressure to effectively do their work for them. As with anybody holding a valuable commodity, you have to learn to look them straight in the eye and deny them their request (**you are in the position of power–they need your help, not vise-versa**), to cut them off from your social circles, and if need be, report them to the Office of Student Life (OSL) to address their

problematic behavior. **Always be willing to break the social contract of decent accomodation if you're interacting with someone who doesn't have your best interest at heart.** This is not to say that you should enjoy turning people down (that's psychotic) but rather you must be able to do so if need be.

c) **Have the utmost respect for the people you are helping.** Even if they are your casual friends, maintain extreme professionalism. **People who admit they need help from you are already in a vulnerable position–it is pure evil to take advantage of that** by putting them in a situation that makes them uncomfortable. Not only will such ruin their week or month, but it also could potentially disincentivize them from seeking help from others for the rest of their time in college. Furthermore, **you will develop an extremely negative reputation among your peers for doing so**, as everyone talks.

d) **Don't just give answers, help students learn to find the answer themselves.** This is the ancient "catch someone a fish vs. teach someone to fish" scenario.

e) **Learn from your students.** If you don't know something, admit it, and have your notes and laptop at the ready to perform research in real time. **Teaching is an excellent vehicle by which to reinforce your own knowledge** as well, so don't go in as a close-minded pseudo-expert pretending to speak the one and only golden truth.

---

# V. Navigating the Institution

### *i) Communication is its Circulation, and Email is its Blood*

**Your inbox should be almost (if not completely) empty at all times.** A frankly unbelievable number of students barely check it, then don't sign up for classes on time, don't get a dorm on campus next year, or don't see that a test date has changed. These same students then go on to complain that they weren't told about these events ahead of time–*yes you were, and more often than not, multiple times over multiple days.* **You are a college student now, check your email minimum thrice a day. Without such, you can and will be left behind by the institution**–SCU's administration isn't going to send someone to knock on your dorm's door to get your attention. You have to intentionally verify that you're on top of everything by regularly checking *the only way* that SCU has to contact you–your inbox.

Furthermore, **take care in how your emails are designed!** Respect your contact's titles (use "Dr. X" or "Professor X"), use full (but concise!) sentences, and highlight the important

information that you want to convey using background colors and bold. If you're contacting a professor, TA, or tutor, **always include the class name that you're referring to in the subject line of your email** (e.g. "[ COEN 79 ] Assignment 2 Clarification"). If you're making a request, always thank the recipient for taking the time to consider your message, and ask them if there are any other contacts you should message in case they aren't the right resource to address your need. Lastly, if you're posing a question, also include a sample set of answers (or even just approaches) that you're unsure about in order to demonstrate that you've at least thought a little bit about the question before reaching out.

### *ii) Advocating for Yourself*

College is a time for you to learn about/practice exercising your ability to stand up for yourself–you will almost certainly get all of the resources that you need in your time at SCU, so long as you are **very proactive** in their acquisition. You should never just sit on your rear and expect things to happen automatically–if you need something, speak up about it immediately to the right people to get results.

To that end, **maintain three core principles in your communications: persistence, politeness, and being data-driven**. If your communications are regarding an issue you have (be it with someone or something), don't give the institution metaphorical ammo to dismiss your concern by coming off as having a trivially selfish motive in your complaint. At the end of my advocation emails, as outlined in the previous section, I always like to add a note both thanking the recipient for reading my message, as well as asking for another contact should they not be the appropriate resource to address my concern.

Lastly, **if something or someone "seems off", gather details about the situation and speak up to the proper authorities ASAP!** This advice ranges from applying to issues with other students, to problems with professors, to toxic party attendees, to sketchy behaviors spotted on/near campus in the surrounding neighborhoods, etc. **The longer you wait to report something, the more likely it is that someone else gets victimized too, and the harder it will be to put a stop to the problematic behavior!**

### *iii) Contacting the Proper Authorities*

The last thing you want to do during an emergency situation is have to run a bunch of high-stress google searches to figure out who in the world can help you–developing a solid understanding of the campus resources at your disposal is critical to avoiding this scenario. I've

outlined some general resources to keep in mind below, all of which you'll be receiving further information about in your inbox as a new student:

a) **Office of Student Life (OSL)**: This office is primarily used to handle issues pertaining to problematic students actively causing (or who have the potential to cause) some sort of harm–be that harm directed to themselves or others. **Always reach out to the OSL sooner rather than later.** SCU is a wonderful school with absolutely zero room for BS, so don't let unacceptable or dangerous behavior fly-by unnoticed.

b) **Cowell**: This office is designed for on-campus physical health services, including support for sports injuries, hormonal concerns, sexual health (so if you're sexually active there is zero reason not to get tested for STDs like a responsible adult), etc. Make sure to try signing up for an appointment prior to attending.

c) **CAPS**: This office provides student mental health services. In the wake of a student tragedy in the late fall of 2021, a lot of anecdotal news was circulating around SCU's student population regarding just how awful CAPS was at their job. With that being said, SCU's administration now insists that they're making changes to improve CAPS, though I personally don't have any data to confirm or deny their effectiveness. **The one tip I do have with CAPS is to make sure you properly communicate the seriousness of your situation when contacting them, so as to avoid potentially dangerous delays in your scheduled meeting.**

d) **Community Facilitators**: Called "CFs" for short, these are SCU's equivalent of RAs in other colleges (they're generally the ones sending emails to you about dorm-based events you can attend). CFs should be the first line of contact you go to when addressing problems with housemates or your dorm community members, as well as to address any infrastructure issues you spot in dorm common areas (e.g. in the laundry room, the kitchen, in elevators, etc.). **Just make sure to treat CFs with respect**–their jobs can be incredibly stressful, and the last thing they need is for a smart ass to mouth off and disobey simple rules that were designed for that smart ass's own safety (pardon my French).

e) **Campus Safety and the Police**: Campus safety should *generally* be the number one resource you exercise in the case of an emergency. Please note that this is only within reason of course–an active overdose or murder should absolutely trigger a 911 call first. However, if someone is robbed, assaulted (be it physically, sexually, or in a verbally threatening/escalating manner), trapped in a building somehow, or in generalized peril, **you need to reach out to Campus Safety IMMEDIATELY and treat them with**

**respect while providing as much data about the incident as possible** (obviously don't stick around in a dangerous situation just to get more details though)! Campus safety will know what questions to ask and how to handle dangerous situations, and if need be, they also have a direct line to the police who they'll contact for you as required. **Every student should have Campus Safety's contact information (which you can find online) in their cell phone at all times.**

f) **Housing and Requesting Work Orders**: If you have an issue with an appliance in your dorm or your apartment's common area, you can email the housing department at SCU for instructions on how to place a work order that'll address the issue (you can find SCU Housing's email online).

g) **Deans and Department Chairs**: These folks should rarely be contacted out of the blue, as they are incredibly busy folks and SCU already has a number of other resources you can use before escalating your isse to such a high level.

   i) Personally, the only circumstance that I've had to reach out to them for was in order to report a professor of concern (who shall obviously remain unnamed) after they employed highly questionable teaching strategies. Upon reporting my concern, the professor was talked to, and I was given an outline of the changes taking place in order to ensure the issues that I raised would be addressed. **Note that this was an extraordinary circumstance however, and you shouldn't be reaching out to the upper administration willy-nilly** every time you get an A- you disagree with–the issues I'd reported had been well known among several generations of students, and hence they made for a solid case that required the upper administration's attention.

   ii) However, **if you do spot a problematic faculty behavior pattern, please make sure to use the upper administration as a resource–don't just brute force your way through the situation until you can move on and forget about the class!** At a $60K/year tuition, you're paying nearly a quarter million dollars for four years at SCU. Demand an education quality that reflects such from your professors! This obviously doesn't mean that you should report classes that are tough because the content is complex, but rather those that are tough because the teacher isn't doing their job properly (something that is blissfully rare at SCU, with the aforementioned professor above being the only instance when I've experienced such in my four years here).

h) **Engineering Peer Advisors and Faculty Advisors**: These are the folks you should always be consulting before you make any significant decisions regarding your engineering course load or schedule, as well as to receive info on academic tutoring services. The earlier you reach out to them the better, and not properly exercising them as a resource risks you doing nearly irreparable harm to your ability to graduate in four years! Whereas a lot of majors have enough free time to afford about a year or so of screwing around, an engineer's schedule is so packed that one or two ill-planned and ill-timed quarters can risk crashing your progress. Further note that **you can switch faculty and/or peer advisors if need be**, just contact Shane Wibeto (the director of engineering undergraduate programs) to request such.

i) **Professors**: Prior to reaching out to professors with a logistical question, make sure to check their online syllabus first–it'll contain a bunch of answers to many of the questions you'd otherwise waste your professor's time with, including when and where office hours are, how is grading done, what's the late work policy, are there make-up exams, when is an assignment due, etc. Also reach out to professors for academic questions in their office hours sooner than later, and contact them regarding your class standing prior to dropping out of a course lest you be mistaken about where your current points place you relative to your peers.

### *iv) Leveraging SCU's Size and Silicon Valley Benefits*

SCU provides a number of benefits to students by right of its size and location. Its smaller student population is such that a sense of community is genuinely fostered between peers, with collaboration being much more prevalent than cut-throat competition. Furthermore, its small class sizes enable you to have an enhanced, individually memorable relationship with each of your professors–something highly beneficial when it comes to getting meaningful recommendation letters from them for your corporate and graduate careers. Additionally, you'll also be empowered to be a hyper-engaged member of the class–if you take the time to ask and answer questions, you'll be able to stand out among your peers and genuinely contribute to the class's overall learning experience (something significantly more difficulty to accomplish in a UC's lecture hall with 300+ students).

SCU's proximity to Silicon Valley isn't anything to scoff at either–I've personally attended multiple events hosted by Google, Facebook, and Microsoft on campus (not to mention those also hosted by many non-trillion dollar companies as well), wherein advice and internal contacts are provided to attendees in order to increase their chances at attaining an internship. **Pay**

**attention to your inbox for invites to attend these events!** At career fairs, a number of representatives from Apple, Amazon, all of the aforementioned presenters, and **so many others** show up in-person to evaluate candidates and answer questions, with each of them having at least some internal SCU contacts in order to ingratiate our university's applicants to their recruiters (Microsoft in particular has recently been **heavily** recruiting top SCU students).

In addition to bringing in top-tier companies, SCU's location is also such that a number of COEN professors have at least had **some** exposure to the Silicon Valley tech scene–whether directly via past and/or present involvement, or indirectly by observing their prior students' progress post-graduation. You can leverage this by sending professors emails with questions or attending their office hours–something that you absolutely ought to do in order to understand any advantages or misconceptions that you hadn't previously appreciated about your industry. Furthermore, irrespective of their direct corporate exposure, I've found that you'll have an amazing slate of COEN professors at your disposal throughout your time at SCU–both with respect to the knowledge and research that they bring to the table as academics, as well as in terms of their quality as lecturers when it comes to effectively communicating inherently complex topics.

---

# VI. Extracurriculars

### *i) Personal Programming Projects*

As mentioned earlier, college is a vehicle by which one is introduced to a variety of topics that they otherwise would've almost certainly never encountered–or at the very least, never encountered so quickly and with so much expertise teaching them right off the bat. With that being said, you won't stand out if you stumble into the job market wielding that which every other applicant already has–surface level knowledge. Corporations aside, this is also true for grad school candidates–**companies and researchers alike want folks who are invested and talented in their craft that stand out from the army of the average.** One of the best ways to do this, as far as I can tell, is to accrue programming experience via extracurricular personal programming projects.

The hardest part of doing a personal programming project is coming up with an idea to code–it could start with something as simple as a tic-tac-toe game, and then get as complicated as an optimizing JIT compiler for a custom programming language. Once you nail that down, the

process of just writing out what code you can, researching to learn more about your tools to tackle what you're stuck on, then writing out more code to apply what you've absorbed, repeating ad infinitum (with a few reach-outs to peers and professors thrown in for good measure) is actually extremely straightforward from a conceptual standpoint. When going into the personal programming projects that I did throughout my undergrad (the vast majority of which can be found **on my GitHub account**), I generally had effectively no clue as to how I would fully implement everything that I wanted to achieve–I simply went in with a dedication to learning about how I could accrue the knowledge required in order to accomplish my goals, and I was willing to iterate upon my approaches until I ultimately achieved what I was looking for.

Personal programming projects served to familiarize me with the process of individual software development and concept research, helped me become intimately familiar with the programming technologies that I used during development (including IDEs, programming languages, debuggers, etc.), made me more open and excited to learn more about increasingly challenging CS topics and paradigms, and ultimately gave me a profound appreciation for the vastly different techniques there are to codifying computations. **I never ended up having to do the "leetcode/hackerrank grind" for technical interviews when seeking internships, as I'd internalized the process of fluidly adapting and composing combinations of algorithms and datastructures into both simple and intricate solutions** to overcome nearly any arbitrary task (at least to the level of complexity that you'd see in a technical coding interview). **You don't need to memorize a bunch of high-level concepts if you can appreciate and adapt their underlying building blocks in order to spawn the high-level concepts yourself as needed.** While programming is initially slow and cumbersome to all newcomers, do it for enough time (and with enough regularity) and you'll find yourself composing code nearly as fluidly as you'd have a verbal conversation.

### *ii) Conducting Academic Research*

Whereas every COEN student under the sun wants to secure a corporate internship, far fewer are enticed by the opportunity to conduct undergraduate academic research. However, I'd put forth that such puts students at a significant disadvantage–**research positions are treasured both within megacorporations as well as in graduate schools, and conducting research makes for an important dimension of the CS universe!** By not performing any research in college, you risk never getting exposed to this world–thereby practically trapping yourself into either choosing to do traditional software development, or forsaking your CS degree altogether (this isn't to say that either of these options are inherently wrong, rather just

that you may find them undesirable–with research being your unknown ideal scenario all along). Personally, I would've almost certainly never gone the grad school route had I not had such a positive experience pursuing academic research as a Junior and Senior.

It's true that traditional corporate internships pay **way** more cash in the short term than research experiences, but **college is a time to invest in your ability to earn–rather than trying to earn immediately!** Getting exposed to undergraduate research is an excellent means by which to cultivate a more diversified background, one that both opens the door to a better understanding of what you want out of your CS degree, as well as enables you to apply for positions in grad schools and companies that you otherwise may have struggled to qualify for! Academic research often requires you to learn to work with your faculty advisor(s), fellow research teammates, publications you'd wish to target, and the vast sea of knowledge which you must traverse (and ultimately contribute to) in order to get your novel work off the ground.

**In order to start getting involved with research**, I'd check out what your school's CS professors are currently working on, both by looking at their personal website, as well as by contacting them directly either in person or via email. Then, once you find a project or concept that interests you, ask them if you could get involved with it as part of their research team (or if they have a related project that you could help out on instead)! You can also take the extremely bold approach of identifying a knowledge gap in the world of CS, then propose a means by which to fill said gap to a would-be faculty advisor in order to get their help when trying to publish your findings (be wary not to reveal too much though, lest your idea be hijacked). Additionally, please also note that while **most students start conducting research as a Junior, I've seen folks as young as freshmen** get involved by contacting a professor to attend (and contribute to) their weekly research team meetings! Furthermore, beyond building up your resume, developing novel skill sets, and giving you valuable CS experiences, **academic research can also often yield you academic credits** to increase your GPA and satisfy certain requirements for graduation.

Having covered the benefits of research, **there are of course still certain things to be extremely wary of** when pursuing such. The main concerns that come to mind are participating in so-called "paper mills", which generally force students to compromise for academic clout at the expense of enhanced understanding, and the risk incurred by working with famous professors:

a) **"Paper mills"** are groups of students that work under a professor with the express aim to produce a high volume of papers (you see this sometimes when a professor is trying

to get tenure), **at the expense of the researchers' quality of learning, life, and academic literature.**

b) Other students accept doing grunt work for a professor's research, not hoping to actually learn anything new, but rather just trying to grasp for scraps of academic clout that they can put on their resume to say they contributed to a well known professor's project. In my opinion, you ought to **avoid this scenario like the plague**: if you're truly smart enough to do research, you're almost certainly too smart to accept this time wasting BS. **Only do research if you're learning from it and are interested in it.**

c) You should also know that **you'll only get as much of your advisor's time for support as your project concerns their future career growth and personal interests**–it's not unheard of that certain well-known professors just give teams 10 minutes a week to help/discuss their project, with that 10 minutes being completely unscheduled and likely to arrive randomly at 10pm one night.

The last point I'll hit on pertaining to research is **the fear of "negative results"** and (by extension) not being able to hit on something big and new in the end–road bumps happen in every project, but the key is to understand why the road bumps are there, and to record everything about your experiences while they're fresh in your mind. That way, when you do finally hit on something after many, many trials and tribulations, you'll be able to outline everything that didn't work, why that was the case, and why what did work in the end actually succeeded–so as to both provide a meaningful result to the world at large, as well as to close off "failure doors" to other researchers who can now understand why some of your initial approaches fell through.

### *iii) Professional and Social Clubs*

Getting involved on campus for engineers typically comes in the form of joining a club or two. At SCU, the way to join a club is to start attending meetings, and the way to leave a club is to stop attending meetings. As far as I'm aware, there is no formal membership status for any of these organizations–they are fluid groups by design that allow for exploration and non-commitmental attendance. With that being said, those that are committed could be eligible to run for a leadership position in the club–something that comes with both great responsibilities and benefits that I'll elaborate upon further below.

Clubs tend to come in two flavors: professional and social. Professional clubs tend to represent academic disciplines (ACM for computing, IEEE for computing and hardware, ASME

for mechanical engineering, etc.), and will strive to host professional development events throughout the year (discipline-specific workshops, resume-building events, hackathons, etc.). Social clubs include groups that cater to specific hobbies and interests, with focusses ranging from gaming to books to sports. While involvement with both of these kinds of clubs is prevalent on campus, just make sure to prioritize your academics over the social scene. If you'd like to attend a workshop on Sunday (even if it's hosted by a professional club) but have two midterms on Monday, have a friend or contact go and take notes for you instead. As a last note, you can find a list of the currently active clubs at SCU by poking around online.

### *iv) Honor Societies*

Honor societies primarily serve as a means by which to officially recognize students with outstanding academic performance (based on their cumulative GPA). While additional benefits may include access to exclusive resources and/or the opportunity to give back to the community, adding the Honor Society's name to your resume is generally the main motivation for people to join. Examples of these at SCU include Tau Beta Pi (the engineering honor society), Alpha Sigma Nu (the Jesuit honor society, though your specific personal faith [or lack thereof] doesn't impact eligibility), Sigma Xi (the scientific research honor society), Upsilon Pi Epsilon (the computer science honor society), and Phi Beta Kappa (the liberal arts and sciences honor society, from which Tau Beta Pi split once PBK became lame [I'm only kidding here {kind of}]). Note that each of these societies at SCU are technically just chapters of international organizations (SCU's Tau Beta Pi chapter is called "California Zeta"), meaning that you're not just getting access to a local organization's resources, but rather those accrued by a nation-wide centuries-old set of organizations.

Additionally, despite having greek-letter names, these societies are **very** unlike fraternities–there's no hazing, they have extremely limited fees (Tau Beta Pi only asks for $100 for lifetime membership, with the fee only being required to pay for chapter materials that you'll be receiving back anyways), and membership is by invite-only (which you'll typically receive via email in your junior or senior year). Should you receive an invitation to join, you'll likely be invited to submit an essay of some sort, as well as be requested to attend an interview or information session. After that, you'll receive the final decision in the coming months as to whether or not you have been officially accepted into the society.

Lastly, something critical to keep in mind with honor societies is that the initiate's character is also frequently taken into consideration when evaluating eligibility. This means that while grades are absolutely important, so is your reputation on campus–especially with

pre-existing members of the society in question. Just because you're the smartest kid around doesn't mean you get to be the chief asshat too, and I've heard stories about potential initiates getting voted out (despite their excellent academic achievements) due to a rotten reputation. Note that such is exceedingly rare, but when it happens, it's tragic–being a decent human being should be the easy part when it comes to getting accepted into these groups.

### *v) Greek Life*

If you read my opinion on the "work hard, play hard" mantra under "Common College Misconceptions" above, you can likely guess what my opinion about fraternities are–namely, **probably don't join them as an engineer**. While I do personally know *some* students that have done ***incredibly*** well in engineering while simultaneously taking part in a social fraternity, I can count them on one hand. I do know, however, **over a dozen engineering students** with academic performances that range from extremely *meh* to *yikes* who've joined frats, a trade-off that I'm not convinced was particularly prudent to make given that tuition at SCU is $60K/year.

If you do insist on joining a frat, look into Theta Tau–it's a "professional" engineering coed fraternity that (at the time of writing) has a decent enough reputation for helping out new engineers, rather than actively sabotaging them like a number of the other frats you'd otherwise join. While I'm not a member of TT, I know a number of high-achieving folks who are, and it seems that they have a solid network of active upperclassmen members to get advice from for classes and career questions (note that TT also doesn't haze its candidates during the rush process to join, which is a massive plus).

### *vi) Leadership Positions*

Another important dimension to extracurriculars is the opportunity to accrue leadership experience–however, only do so if you are ready to take on the full responsibility implied by such! **Do not become a leader for the social, institutional, or career clout and advantages–become a leader because you want to (and can!) lead an organization into a better future.** I've seen terrible leaders get elected before, and they just end up getting kicked out with a horrible, crumbling reputation left of them among their peers after having wasted everyone's time. If you think that not having leadership experience on your resume is bad enough to warrant risking applying to positions that you're not able to fully execute, I can promise you that (from first-hand experience observing such) having a bad leadership experience is infinitely worse.

This (of course) is not to say that you shouldn't be bold and strive to apply for such positions–rather, don't go for it if you already have a ton on your plate, or if you're only interested in the title. **Being a leader requires time, organizational, and emotional investments into the success of your organization**–if you can't provide that, then you have no business taking charge of an org.

With that being said, **becoming a leader can make for a fantastic experience!** You'll learn to coordinate with a board of fellow club officers, oversee the members of your org, host a variety of events, coordinate with SCU as an institution for funding and/or general support, make industry connections in order to have them meet your club members, garner attention to your org via marketing, and more :) Depending on the exact kind of leadership you're taking on, you may even have the ability to interview and hire people–something that makes for an excellent learning experience when it comes to navigating a wide variety of social dynamics (I found this to be especially true for me as the lead engineering peer advisor).

In terms of leadership advice, everyone has a different style. Having lead both Tau Beta Pi and the Engineering Peer Advisors this past year though, I do have a few general tips:

a) **Don't underestimate the power of just showing up.** Shockingly few people actually try applying for leadership positions–just putting in your application is 80% of the work to get elected.
b) **Follow your gut.** If you feel uncomfortable about someone or something in your midst as a leader, investigate why feel that way–depending on the organization that you're leading, you may not be able to afford letting someone in (be it for an event, as a member, or as a fellow officer) if you're not 100% sure that they have good intentions in mind. This also applies to ideas–if you think something's risky, 99 times out of 100 it's better to play it safe.
c) **Be extremely organized.** You should be treating your leadership position as if it were another class–keep a schedule, set reminders, do your homework, coordinate with others, etc.
d) **Listen to others for advice.** It's important to keep in contact with your fellow officers and the org's members in order to verify that you're doing a good job. Furthermore, while it should be obvious, actually make sure to take their advice to heart–***being a leader is ultimately a position of service, and you must make sure that you're properly serving your members without jeopardizing the future of the org.***
e) **Keep org reflections.** Document what did and didn't work under different circumstances, and pass that data down onto the next set of leaders after you. In a

mirrored fashion, make sure to also learn from the mistakes and successes of your predecessors, so as to avoid repeating disasters and unoptimized approaches.

f) **Have a good time :)** Don't let leading only stress you out, enjoy the position as you use your influence to spread good throughout your org's members. They'll remember you in an enhanced light throughout their time at college, so make sure you leave a lasting, positive impression on them.

g) **Lead by example.** No one (not even yourself) is going to take you seriously if you espouse A but live by B. Keep it real, and be the model org member yourself as you lead the others.

---

# VII. Corporate Internships

### *i) Things to Keep in Mind*

Nothing has reminded me more of the process of applying to undergraduate schools than the process of applying to corporate internships–there is a bunch of crap floating around regarding what magic spell you have to cast when and where in order to catch the eye of our corporate overlords, but the reality is that the entire process is actually pretty straightforward.

a) **Companies are generally terrified of freshmen and sophomores.** This is in part justified, mostly because the majority of these students hardly know a lick of real-world applicable CS. However, that fear does undeniably suck for those underclassmen with true potential, though I do have advice for these folks in the section below.

b) **Genuine technical knowledge is vital.** Don't try and memorize your way into a position, you will be sniffed out and expunged accordingly. **The leet-code/hackerrank grind is absolutely atrocious and an insult to a good programmer's intelligence.** However, you actually (thankfully) won't need to do it (much) if you've accrued a solid sense for data structures, algorithms, and their application to specific use cases via personal projects and prior experience.

c) **You've got to apply, apply, apply!** Think in terms of the several dozen–internship applications are a numbers game, you can't apply to just 5 or 10 of them and expect a good shot (*especially* if those 5-10 are top-tier tech companies!). Additionally, **don't be discouraged by rejections, or even "failed" interviews!** Learn from what you can about the experience, but fundamentally understand that these opportunities are highly

competitive, so initial failure is to be expected! The key is to not get disheartened: keep your head held high and don't stop searching for opportunities!

d) **Apply early!** If you want an internship for the summer of year N, you should be applying in the late summer of year N-1. Most top tech companies have their set of interns fleshed out by October/November, so you **have** to be proactive about applying to positions as early as possible!

e) **Don't shy from leveraging contacts.** There is undeniably pride in being able to score an internship all on your own, and this is in fact an incredibly useful experience to go through at some point in your career. However, especially at the start of your internship journey, family and friend connections to corporations can lend you an immense amount of help in order to get to the interview phase (note that such is generally the extent to which your contacts can help you–you won't get a free ride to a position, rather the referral will just help you get past the company's initial screening phase).

f) **Honesty is everything.** Lying on your resume will not only lead you to almost certainly being caught, but it will also tarnish your reputation henceforth at that company. **People have been fired from their positions at companies after it came to light that they lied about their credentials to get the job–*even if they were a great employee after they got in!*** The risk factor is just too high for corporations that the disingenuous attitude held by a lying employee could seep its way into their future endeavors. Furthermore, **if you don't know something about a question that comes up in an interview, you're infinitely better off coming clean!** Few things come off as dumb as: "Can you do X?" "Yes!" "Ok, how does it work?" "Uh…".

    i) For the record, I consider doing things like adding "ARM Assembly" to the list of programming languages you know after taking COEN 20 to be a lie. Could you write quicksort in assembly right now if I asked you to, even if I showed you the pseudocode for quicksort? Probably not. If you want to include a variety of languages or technologies on your resume, just put the amount of time you've been using it for in parenthesis afterwards to provide a rough estimation of your competence (e.g. "C/C++ (3 years), Java (6 months)"). The last thing you want is to say you're good at something on your resume, only to fumble at it in a basic example during an interview.

g) **Your interviewer is there to help you!** If the company is hiring, they want someone to join their team! Smile and be polite, and verbalize all of your problem-solving approaches for coding questions in order to demonstrate your ability to think (and apply prior

knowledge) on the fly! The technical interview is more so about testing your abilities and intellectual rigor as an engineer than it is to literally solve the problem on the board. Ask your interviewer gentle questions and clarifications about corner-cases (it's expected for you to do so, and a red flag if you don't), and incorporate them into your answer! Lastly, focus on trying to get a working solution up and running prior to optimizing your approach, but **do** explain the trade offs you're making and what that implies for the time/space complexity of your code.

h) **Return offers are never guaranteed.** Don't bank on getting grandfathered in trivially after an internship, as your time there effectively serves as an extended interview session between you and the company. Note that this goes both ways: you should be using such to evaluate the corporation too (though beware that, as an intern, you're likely to see everything through rose-tinted glasses). Furthermore, don't just work to accomplish the MVP (minimum viable product) that they set out for you–**a valuable intern seeking a return offer will work with their team in order to learn from their experiences and create a product that far exceeds expectations.**

### *ii) Specific Advice for Underclassmen*

Freshmen and sophomores are in a uniquely tough position when it comes to applying to corporate internships (due to the aforementioned apprehensive disposition most companies have towards them), however, there are a few things that underclassmen can do to increase their chances (or at least use the time they would've spent at an internship productively to increase their chances next time around):

a) **Don't fear small companies!** A lot of first years want to go from zero to Google just 14 days deep into their bachelor's. However, internships are actually typically designed as the first step in the hiring process of college students, so any internship you get in your first two years won't be a "real" internship (e.g. you won't be working on actual products). For reference, one Facebook employee flat out admitted to SCU students that their underclassmen internships are basically just glorified hackathons. Instead, apply to smaller (even start up!) companies where you'll both be more likely to get accepted, as well as be more likely to have at least a bit of real, hands-on experience. Note that you likely won't get paid at first, but you're a freshman! The goal is to get experience and education at this point, you likely don't have the skills required to demand a high salary yet.

b) **Apply early in fall!** As mentioned earlier, the optimal time to apply to internships is near the end of the summer prior to the summer for which you'd like to intern. Many freshmen are caught off guard by this when they come to college in the fall, so if you are going to apply, you best do so ASAP as soon as you arrive on campus.
c) **Alternative: develop your skills and do personal projects!** If you don't get into an internship, that's not an excuse to slack off during the summer. Teach yourself a new programming language or set of technologies by doing a personal programming project that you can put on your resume! The summer is a bunch of free time–something exceedingly rare in the life of an engaged undergrad engineer. Capitalize on that free time by using it to develop desirable skills!
d) **Alternative: pursue academic research!** A few students that I know also pursued research opportunities over the summer as underclassmen–this can be done by approaching professors and asking if they have any openings for you to do such. However, be careful to make sure that the research area actually interests you, otherwise it'll be a hellish slog of reading dozens of papers you couldn't care less about. If you do care about the area though, it'll be a valuable experience in learning about the most recent, cutting-edge developments in the area, as well as an opportunity to learn how to process enormous amounts of data from papers relatively quickly.

---

# IIX. Grad Schools

### *i) Why Go Grad?*

Going to graduate school is a deeply personal choice, and one that I can't even be sure I'm doing correctly given that I'm still a senior in undergrad. With that being said, I'll still articulate my own personal reasons for doing so below:

a) **It's not for more money.** Unlike some other majors, a graduate education is wholly unnecessary if hard cash is your sole goal as a COEN. Something especially true for those aspiring to be professors, you'll undeniably make heaps of more money by going straight into industry.
b) **It's not for the title.** I can hardly think of a dumber reason to pursue a PhD.
c) **I love to discover knowledge and be around people obsessed with CS.** Note that both of these can also be done at corporations, but they're especially straightforward to

do in grad school. Being that grad school isn't required for higher earning power as a COEN, its members are (theoretically) self-selected to be those crazy enough about CS to bother pursuing a grad education in it. Furthermore, getting into research is significantly easier and expedited in grad school than from an entry position in the corporate world, so if the cutting edge calls to you, grad school is the place to be.

d) **Good Professors and TAs make all the difference in the world.** While I'm still a bit on the fence regarding academia vs corporate research, I do genuinely enjoy teaching others, and would like to have a positive impact on a number of students' lives. Having had amazing, mediocre, and truly *meh* professors, I know that having a great educator makes for an incredible experience. I'd be honored to be able to provide that back to folks in one way or another, be it either by actually becoming a college professor, or by educating others via a different avenue.
e) **It can open the door to more opportunities.** Given that a Bachelor's is a subset of a Master's or Doctorate, it's not too surprising that such a relationship is reflected on the job market. While you'll always be able to apply to entry-level positions, getting a grad education can open the door to more research opportunities in both academia and the corporate world. Further note that some may insist that the PhD was a waste of time if you only end up applying to entry-level positions anyways, but to me this only rings true *if you didn't join a PhD for personal and intellectual fulfillment* and instead only aimed to have it boost your career–which, while undoubtedly a nice side effect if it happens, shouldn't (in my opinion) be the driving factor behind getting your grad degree anyways.
f) **Grad schools are very straightforward to apply to, and have more means to support you financially than in undergrad.** *Note that ease of entry isn't an inherently good reason in and of itself to pursue any avenue in life, however, I will take this opportunity to clarify just how easy it is to apply to grad schools relative to undergrad in order to clear up any misconceptions that students may have.* Some folks have a bit of PTSD from the nightmare that is applying to undergrad institutions, however, applying for graduate positions is much more straightforward. For PhDs, I generally only had to write one generic essay that applies to every school (a so-called "statement of intent"), the last paragraph of which I customized to each institution that I was submitting the essay to (mentioning specific research programs/projects of theirs that appealed to me, specific professors that I'd like to work with, etc.). Furthermore, an increasingly high number of CS graduate programs are no longer requiring the GRE (a position that is frankly long past overdue), meaning that you won't have to worry about SAT/ACT flashbacks. Lastly,

you can also get more financial support as a graduate student by TAing for classes, and/or getting funding from your advisor if you're in a PhD program.

### *ii) Application Advice*

While there's obviously no magic formula for getting into a specific grad school 10/10 times, I'll still do my best to outline some of the common knowledge that I've accrued in my time researching the process online and by talking to a variety of professors:

a) **Your undergrad GPA matters.** Unlike ***most*** of the corporate world, GPA is going to play a significant factor in your ability to get through the first few rounds of grad school application screenings. If you have a cumulative GPA below 3.0 (or below sub 3.5 if I'm being realistic), it'll almost certainly be incredibly difficult to stand out among the sea of excelling students that you're competing with.
    i) Anecdotally, when I applied to grad schools with a 3.96/4.00 GPA, I was told by one professor (during their applicant interview process) that such was high enough to nearly guarantee me a spot in the university's PhD program.

b) **Undergrad research experience is a big plus.** Note that this research doesn't necessarily have to be in the domain that you want to focus on in your grad education, but being able to articulate what you learned from (and liked about) the experience of analyzing a topic, producing new knowledge related to it, then writing and publishing a paper about it is of great aid when it comes to expressing your value and dedication to grad schools.

c) **Personal projects can help you stand out.** If you're the kind of person who likes to code for fun while diving into more advanced areas of CS in their free time, I've been told that grad school is almost certainly for you. For example, while I wanted to do a PhD in Programming Languages (PL) research, by the time that I applied, I'd only done undergrad research on the viability of electric vehicle to grid (V2G) technologies. However, the vast majority of my personal programming projects did pertain to PL (I love writing interpreters, compilers, custom object/macro systems, and more), and hence I was able to demonstrate my personal interest in (and dedication to) PL as a field to my potential PhD advisors. Furthermore, despite being in a different industry, I was also still able to communicate the generalized lessons about research that I took away from my time spent publishing a V2G paper.

d) **Good relationships with your undergrad professors are vital!** You'll be needing 3 recommendation letters to apply to grad school, ideally with all of them coming from

professors. For mine, I chose my faculty advisor (who, having been with me from the start of my freshman year, knew how solid I was across all of my classes), the professor I did research with (who could attest to my research abilities and dedication), as well as my compilers professor (who worked in a domain similar to that of the area I was interested in, and in who's classes I had excellent performance).

e) **You need to project strong grad motivations and potential.** You can't afford to come off half-invested in research and/or teaching (ideally you're interested in both) when you write your statement of intent–make sure to have a rock-solid set of motivations that you can clearly articulate as to why you should be invited to attend their specific grad school.
f) **Keep it real and recent in your application.** Grad schools get the "I've been coding since 2nd grade"/"I'm literally Einstein" spiel far too often. For one thing, they don't believe you. Secondly, they don't care if you've been a prodigy all your life–they want to know about what you are doing **right now** that makes you an excellent candidate for their graduate program. The last thing you want to do is be disingenuous or embellish the truth in your application, for having such come to light will destroy your reputation.
g) **Apply earlier rather than later!** The actual application process is shorter than you likely think, but it's incredibly easy to have deadlines sneak up on you. Just bang them out as soon as you have all the material you need ready for your essay! Also, **make sure to give your professors at least one month to finish writing your recommendation letters!**
h) **Sometimes the perfect PhD applicant can't be let in.** There is one fundamental aspect of pursuing a PhD that is completely out of your control, and that's the research team's funding situation. Even if you have everything that the team would traditionally look for, their grants can only support so many students–meaning that, due to the whimsy of budget constraints, there's no way to guarantee that even a "perfect applicant" gets in.

---

# IX. Staying Safe in College

### *i) Follow Your Gut*

While in my experience SCU is generally composed of decent and well-intentioned people, like any reasonably large institution made up of human beings, it does undeniably have

some wolves among the sheep. **If you ever think that someone or something in a situation "seems off", play it safe and leave the area *immediately*. Forget social decency–your own well being trumps the need to be polite to someone who has rotten intentions in mind.** Contact the proper authorities (campus safety and/or 911) **ASAP** and provide them with as many details as possible, though (of course) **never** stick around a dangerous situation just to gather more data as your own personal safety is paramount!

### *ii) Party Smart, Not Hard*

While regularly partying is generally ill-advised for COENs (if that's your thing you should almost certainly pick another major and/or genuinely reevaluate whether college is worth attending for you), the reality is that some folks will still go out. Furthermore, because a non trivial number of engineers who attend such may have never done so before, it's all the more important to be careful when you go out.

a) **Never walk home alone, and only walk home with true friends you know from school.** There's no guarantee that a random acquaintance is well-intentioned, or even that they're from SCU (outsiders sometimes sneak their way into these things). If your friend leaves early, go with them–you're always better off safe than sorry.
b) **In 2021's fall quarter, there were multiple rapes and even a death associated with frat parties**–surrounding yourself with people you trust (at least one of whom is ideally sober) is absolutely essential to avoid potential trauma.
c) **Never drink from pre-opened canisters, and never put your drink down.** You have no idea who put what where and when, and that initially "nice guy" you just met could turn out to be a sociopath fueled by adrenaline, alcohol, and testosterone.
d) **Ideally don't use any substances at all, but if you do, bring your own to the party and don't take things you find while you're there.** Pill presses are a thing that exist–just because something looks like a Xanax doesn't mean you actually have a clue as to what the drug is. This also applies to substances from supposedly "trusted sources"–just because you trust person X not to intentionally hurt you doesn't mean that you can also trust who person X got their supply from.
e) **You can always take more, you can never take less.** In other words: "pace yourself". Once you consume a substance, it becomes either impossible or nearly impossible to reverse the effects of your decision–do your research and take breaks.

i) For example, **did you know that it takes 30-90 minutes to feel the full effects of alcohol?** Take your time between drinks–drinking till you're drunk is how you end up dead or in the ER.

f) **Driving (or riding with a driver) under the influence of anything is insane.** I don't care how many people say that they "drive better high", **don't do this.** Putting aside the legal liability you assume by doing so, we live in the age of Uber and Lyft–there's absolutely no excuse not to get a ride home from someone sober.

### *iii) Avoid the Avoidable*

**Take basic common-sense measures to protect yourself.** Don't wear flashy things that attract the attention of robbers, don't leave windows open while you're gone or changing clothes, lock your car, don't interact with the homeless as you have absolutely no way of knowing how mentally stable they are (if you want to help them, contact the proper authorities who are trained to do so), don't engage in conversations with strangers (especially if they walk up to you out of nowhere), always check the license plates of Uber/Lyft cars to verify you're getting in the right vehicle, don't leave your backpack or laptop alone in a public area if you ever want to see them again, always keep your phone charged in case you need to make an emergency call, etc. and etc.

Another key thing to keep in mind is to **prefer walking within campus instead of around campus**–there have been a few **rare** instances where someone is almost robbed or gets flashed by some creep on the roads around the area. However, these bastards typically never go on campus, so especially if it's late at night, prefer navigating through the heart of SCU rather than around its border.

### *iv) Address Health Concerns Sooner than Later*

Note that this applies to both physical and mental health–if you are worried about the integrity of either, **the swift utilization of resources at your disposal is critical to ensure the prevention of further damage and the mitigation of current risks.** Whether it be a sports injury, potential concussion, hormonal concerns, pregnancy scare, STD anxieties, or anything pertaining to your mental well being, you need to get an appointment with the Cowell health center (for physical health) or CAPS (for mental health) ASAP! You don't want to play around with this stuff, or wait while crossing your fingers for the issue to go away on its own–**your own health must alway take precedence over your scholastic, career, and social obligations.**

# X. Personal Advice

## *i) Your Reputation Matters*

**In conjunction with your abilities, your reputation will determine what opportunities are extended to you by institutions, course professors, and your peers.** Raw technical competency will only take you so far–you won't be able to fully capitalize on everything that college has to offer without also having excellent character. Consequently, if you cultivate a negative reputation, you'll be significantly hindering your own academic, career, and social prospects in life! I've seen a few students (blessedly rare as they are) that were shut out of incredibly fruitful opportunities at SCU due to their toxic behavior's reputation.

Additionally, always remember that word spreads about people in college: if you're the kind of person who helps others in and out of class, takes their studies seriously, and is fun to be around, you'll eventually be known for such around campus. Conversely, if you're the kind of person who derides and degrades others, tries to cheat their way through classes, and can't take a joke or lacks basic self-awareness, you'll be shut out and abandoned by those around you. No matter how successful you are, always keep in mind that, as a college student, everyone is fundamentally in a similar (if not the same) boat–no one likes an arrogant jackass, and being cynical about your peers only serves to push potential companions away.

Further note that I believe one's reputation largely stems from the consequences of the intentions that they harbor towards others–I once knew somebody who genuinely stated (with no sarcasm whatsoever) that they wanted to be **the only person** who got an A in a certain class. Not only that, but they were bragging about this goal to those around them–what kind of message could this possibly be sending to their peers, besides shouting from the rooftops "I'm deeply insecure and manifest my fears of inadequacy by degrading those around me"? Wanting to get an A is highly commendable, but only if you do so either for your own satisfaction, or even better yet in order to also be able to help other people with the knowledge you've acquired! Actively wanting everyone else to fail is a cancerous mindset that's thankfully extremely rare at SCU, which makes it stand out as all the more crazy when it does crop up. **Any student (no smarts required) can cut throats to get to "the top"–the only impressive part worth respecting is when someone achieves what they desire while also bringing everyone else along with them to the pinnacles of success!** And in doing so, you'll attain the best reputation you could ask for from your peers.

### *ii) Be Proactive*

Staying on top of the ball is critical for college (and lifelong) success–a number of the opportunities afforded to you both on and off campus will be time sensitive, and **the concept of "first-mover advantage" is a very real thing!** As such, **only strategically delay tasks or communications** (be it to gather more data, wait for an event to occur beforehand, etc.). However, if the only reason you haven't done something yet is "IDK" or "meh", you should almost certainly just start it sooner than later and get the task over with.

With that being said, once your proactivity **has** yielded an opportunity for you that **isn't** time-sensitive anymore, **there can be immense value in keeping your options open!** If you have until a certain deadline to accept a job offer or grad school opportunity, take your time prior responding to such lest other factors come into play that affect your decision (*just make sure **not** to miss the deadline!*). As I'm sure you know, life can throw curve balls completely out of the blue at you–learning to leave room for you to pivot in and out of a situation is a skill worth cultivating, part of which entails not unnecessarily locking yourself in or out of particular scenarios before you have to make a final decision.

### *iii) Collaborate With Others*

**Make an effort to surround yourself with excellent peers**–people that inspire you and give you hope for the next generation of engineers, people that you'd be honored to work with and can see doing great things in the future! Note that this obviously does **not** mean you should shun those that need help, or those that aren't inherently academic/technical rock stars **yet**–rather that you should always keep at least some impressive folks around that push you to be the best version of yourself, so that you can perform the same function for them in a mutually recursive and beneficial way.

Furthermore, make sure to appreciate all of your peers! Everyone has value in their own way, and just because someone isn't one of those aforementioned rock stars today doesn't preclude them from becoming such tomorrow. The last thing you want is for people that become successful to remember you as a snotty jerk with your own head so far up your ass that you lost sight of your common humanity. **Have empathy for those around you**–while some folks undeniably have poor results due to laziness, you never truly know what's going on in someone's life. They could be going through an absolute living hell, and by being nice to them, you're giving them the strength to keep persevering for another day–even if they never explicitly thank you for it.

### *iv) Set Yourself Up for Success*

Consciously make choices in life that set you up for long-term success. Understand your motivations for pursuing something, then set the wheels in motion in order to break down and execute the steps involved to accomplish that thing. Should you run into a situation where you find yourself frustrated at an intermediary step (this was especially the case for me when reading 50+ pages at once for certain CORE classes), **I've found the ability to consciously restructure my short-term value hierarchy to be extremely useful.** What I mean by this is you have to be able to zoom out in your mind to see the broader picture–just like a C pointer is used to indirectly access a value in computer memory, temporary struggle can be used to indirectly access success and satisfaction in a future broader goal. I won't lie, there were times where the work I was asked to do made me want to put my head through the wall–but keeping my eye on the ball by consciously appreciating why I was putting in such effort helped me crush each and every one of those tasks.

With all that being said, one of the biggest challenges in life is that of simultaneously setting yourself up for success, while also not shying away from valuable risks worth taking–it's incredibly easy to succumb to paralysis by analysis, or to simply hide away from the world in a "comfort zone" that only serves to inhibit you from reaching your full potential. One technique that I've employed to circumvent this is to **reevaluate what I consider to be "not success"**: for example, say that there's an election for a student organization coming up, but you're nervous to join in because you think you might not get elected, and the thought of that risk deters you from entering the race. The truth is, not getting elected to such a club is a non-risk: the downside is extremely minimal so long as you learn from the situation and don't harbor unhealthy resentment towards the victor. If anything, it'll show your dedication to the group in question, thereby making you an even stronger candidate next time that elections come around! This thinking also applies to a number of other social situations as well, including making friends, joining clubs, participating in competitions, proposing study groups, asking for help, etc. It's all about being willing to overcome the very human anxiety of putting yourself in an uncomfortable social situation–so long as you do so conscientiously, with genuinely good intentions in mind, and politely, you'll be completely fine and learn a ton about both yourself and other folks.

The last point that I'll make in this paper is to **know your worth, and beware of selling out to your own insecurities.** Note that I'm not saying you shouldn't be making prudent life decisions informed by rational concerns, but rather that if you genuinely kick ass, recognize it and capitalize on it–squeeze life for everything it's got, and don't be afraid of taking an

unconventional path. Not everything is for everyone, but if you play your cards right, you'll be absolutely astounded by what you can achieve in this world–especially if you're the kind of person that makes it to the end of this kind of paper ;)