DIGITAL & COMPUTATIONAL MEDIA TRAVEL TO
COPENHAGEN

[ HACK WEEK ] STUDENTS PREPARE FOR A WEEK OF HACKING
[ LACK OF SCIENCE IN CS ] TORONTO PROF. SPEAKS WITH THE FIREWALL
AT A GLANCE

COPENHAGEN
Learn more about the Copenhagen ITU exchange, one of Georgia Tech’s newest study abroad programs. Interested in media? Design? Games? Denmark? Study in one of Scandinavia’s hippest cities, and experience another culture as you learn about digital media and design.

PAGE 14

IPAD TIPS THE SCALE
One student’s thoughts on how the iPad may be a step in evolutionary scale of computers, and how such a tablet could be used in a classroom setting.

PAGE 7

THE FIREWALL TEAM

PRESIDENT
Karthik Narayan

EDITOR-IN-CHIEF
Chris Russell

CHIEF COPY EDITOR
Aurel Lazar

CHIEF LAYOUT EDITOR
Terris Johnson

CHIEF WEB DEVELOPER
Joy Buolamwini

STAFF WRITERS
Jeannie Choi
Cristina Gonzalez
Ashish Narasimham
Connie Chen
Andrew Panfel

COMMUNITY
CONTRIBUTORS
Gaurav Mathur
Stephanie Tumlinson
Moinak Bandyopadhyay
Aaron Sun
Chaitanya Adgaonkar

SPECIAL THANKS
Meredith Goodman
Tom Pilsch
Cedric Stallworth
AN OPEN INVITATION

KARTHIK NARAYAN
THE FIREWALL PRESIDENT

Hello, students of the College of Computing! I would like to present you the March issue of the FIREWALL on behalf of our team. This issue features articles ranging from hack week to studying abroad in Copenhagen to interesting theory courses.

We are currently looking to recruit new members to our organization, in the areas of writing, copy editing, graphic design, layout, and publicity, so if you’re interested, apply via our website, at www.gtfirewall.com. You’ll get a chance to work with some of the brightest students in the College of Computing and learn a variety of new skills while at it!

The FIREWALL team has more waiting for you this semester, so buckle up your seatbelts, sit tight, and enjoy the March issue.

WANT TO HELP SPREAD THE FIRE?

YOU CAN CONTRIBUTE TO THE COLLEGE OF COMPUTING STUDENT COMMUNITY!

VISIT OUR WEBSITE -- WWW.GTFIREWALL.COM

Writing
Layout Editing
Web Development
Illustrating
Art/Graphic Design
Photography
Sponsorship
Marketing/Publicity
Printing and Distribution

Experience is welcome but by no means necessary!
WHERE'S THE SCIENCE?

U OF T PROFESSOR GREG WILSON TALKS COMPUTER SCIENCE

BY CHRIS RUSSELL

On Friday Feb. 19th, students in Bob Waters’ Objects and Design class were given the chance to attend a guest lecture by Greg Wilson, an Assistant Professor of Computer Science at the University of Toronto. While guest lectures are by no means an uncommon occurrence in the College of Computing, Wilson’s talk raised a question applicable to a larger-than-normal percentage of College of Computing students: if we’re computer scientists, where’s all the science?

Walk through the Commons any day of the week, and odds are good that there will be at least one ongoing argument over what language, IDE, or operating system is the “best.” The problem is, how much of the evidence in those arguments would pass muster for a scientific study and how much of it is just hot air? In medicine, before a company can make any kind of claim about a new drug, it has to go through months of rigorous double-blind studies, but no such requirement exists for many areas of computer science, particularly in software engineering. Often, all it takes to convince an audience of a wild claim is to preface it with, “My friend Jim said,” or, “Everyone knows that,” and the speaker’s word is taken as law.

After his talk, THE FIREWALL caught up with Wilson and asked him a bit more about his ideas and how they affected students.

FOR STUDENTS WHO DIDN’T ATTEND THE TALK, IN YOUR OWN WORDS, WHAT WAS YOUR TALK ABOUT?

The need for more scientific rigor in software engineering. All too often, people in both academia and industry claim that X is better than Y based solely on some personal experience and a couple of anecdotes. We wouldn’t accept that as proof in medicine or business (well, a lot of people would, but they shouldn’t); those of us who build software for a living should have equally high standards.

YOU TALKED A GREAT DEAL ABOUT THE LACK OF BASIC SCIENCE IN MUCH OF

WWW.GTFIREWALL.COM
COMPUTER SCIENCE. WHAT DO YOU MEAN BY THIS?
I don’t know about Computer Science as a whole, but there’s certainly not enough “science” in software engineering. Take UML, for example: where is the evidence that drawing class diagrams and sequence diagrams actually results in better or cheaper software? Or arguments over Java versus Python versus whatever your favorite language is today: we’ve all heard people yell about this, but where’s the evidence? Note that I’m not saying different languages don’t matter---I’m saying that if people as smart and as talented as Larry Wall (Perl), Joshua Bloch (Java), and Simon Peyton Jones (Haskell) disagree, it’s time for us to stop yelling and start being systematic.

SINCE IT WILL LIKELY BE YEARS BEFORE STUDENTS AND NEW GRADS ARE IN A POSITION TO MAKE BIG DECISIONS ON HOW TO CHANGE THIS --

WHAT CAN WE DO NOW AND AFTER WE GRADUATE TO NUDGE THINGS IN THE RIGHT DIRECTION?
You can start by reading Robert Glass’s excellent little book, Facts and Fallacies of Software Engineering, then get the second edition of McConnell’s equally excellent (but much larger) Code Complete. Between them, they’ll tell you most of what we actually know about building software. More importantly, they’ll introduce you to the studies that have been done, so that the next time someone says, “X is better than Y,” you’ll know to ask them for a citation.

HOW DO YOU THINK THESE PROBLEMS AFFECT OUR INDUSTRY? HOW WILL THEY AFFECT OUR INDUSTRY IN THE FUTURE?
You can expect to live 25-30 years longer than your great-grandparents. Most of that gain is a result of us learning to treat medicine as a branch of applied science. I think we’ll reap similar benefits from saying “Prove it!” more often in software engineering.

WHAT ARE PEOPLE IN ACADEMIA AND INDUSTRY DOING TO REMEDY THIS PROBLEM, AND HOW MUCH OF AN EFFECT IS IT HAVING?
A lot of software engineering researchers are now doing this kind of empirical work. It hasn’t had a lot of impact yet, but I think we’ll see that change dramatically in the next 3-5 years.

COULD YOU TALK MORE ABOUT WHAT YOUR BOOKS ARE ABOUT AND HOW THEY TIE INTO THIS TOPIC?
I’m editing a new collection, which O’Reilly will publish this summer, that brings together a lot of the primary research in evidence-based software engineering. We’re still trying to decide on a title, but you can follow our progress at http://third-bit.com.
BY KARTHIK NARAYAN
Of the many computer science courses taught at Georgia Tech, perhaps one of the best is the CS 1050X course. Taught by the lively Dr. Vijay Vazirani, the course is among the most anticipated and popular courses. Several students, including myself, deferred taking CS 1050 to later semesters just to take the section offered by Dr. Vazirani. With only around thirty slots available and over sixty students attempting to sign up for the course, students often snipe Oscar every five minutes to check if a spot opens or if their overrides are processed. Even after the first week passes, several new faces appear in class. When these students are asked if they are new, they reply that they want to just sit in on the class and see what it is like.

Such is the fame of this course. So, what makes this course tick?

No course can thrive in popularity without that zing and oomph. True, it’s difficult to lay a finger on exactly what these aspects are, but Dr. Vazirani has been able to harness CS 1050’s sparkle and turn it into something else.

“BY CREATING A CONVERSATIONAL ATMOSPHERE... STUDENTS TAKE A TEAM-BASED PROBLEM-SOLVING APPROACH.”

Constantly cheerful and encouraging students to do their best, Dr. Vazirani keeps the classroom entertained. Gone are the old ways of teaching where a professor lectures in a monotone voice by drawing dull diagrams and giving PowerPoint presentations. According to Abhishek Shroff, a CS 1050X alum, “[Dr. Vazirani] made sure that no student was left behind in the course, while at the same time, ensured that the... ones with more background knowledge had enough to think about.”

In class, Dr. Vazirani usually introduces a concept, which takes about thirty minutes of the hour and a half class. The remaining hour is devoted to a problem solving session. By creating a conversational atmosphere where students take a team-based problem solving approach, Dr. Vazirani drives in what the real work in the industry is like. A series of problems related to the concept is usually described, and the class is challenged to arrive at a solution.

As many jobs are starting to outsource to foreign countries, the demand for the skills which Dr. Vazirani offers is very high. Michael Qin, another alum, said, “I believe it is classes like Dr. Vazirani’s that can reverse this trend... [the course] is key to ensuring that the best students are not subjugated to slower-paced [curricula].”

Patrick Dillon, a student who took CS 1050X in the spring of 2009, said that Dr. Vazirani’s methods of teaching “allowed the student[s] to speak their minds as he provided slight nudges and insights to the key aspects of the problem[s].” At the surface, these problems seem to be superficial. However, as the students solve the problems, they discover gilded treasure troves, where all sorts of seemingly unrelated concepts suddenly coalesce. By building bridges across various fields, Dr. Vazirani is able to bring out the true beauty in mathematics and convey this message to students, an exceptionally rare quality that I haven’t seen in any other professor or teacher in my career as a student.

Dillon further said that he “firmly believes that the reasoning skills students learn in Dr. Vazirani’s classes will continue to be of use for the rest of their lives, either directly applied or as a firm foundation to build upon.” In fact, the classes of 2008 and 2009 have both had students attend the prestigious BOLD internship that Google offers. For now, it looks like the course has taken a brief lapse in the course listing schedule, but we look forward to seeing it back on track next year!
AND THEN THERE WAS IPAD

HOW THE IPAD WILL CHANGE LIFE, UNIVERSE, EVERYTHING

BY ANDREW PANFEL

As is customary with Apple product launches, Steve Jobs presented the keynote introduction for the iPad amidst cheers and applause from the audience. Answering the question of where a tablet device fits, not in your pocket, he introduced it as a middle-ground category device, intended to bridge the gap between mobile and personal computers. He stated that in order to be that in-between bridge, you have to be better at some things, like web browsing, than with devices in either category. Jobs then sat down in a big chair and browsed the internet. Leaning back with a 10 inch screen just a couple of inches from his face, he smiled and proceeded to tell the audience that the best web browsing experience had arrived.

Personally, I love tablets. My current primary computer is a convertible-style tablet which I bought 2 years ago. I purchased it when I became a student at Tech, and it runs Windows. I take all my notes on it, because it uses a Wacom pen or a finger as input directly to the screen. Writing my notes on it is a digital revolution equivalent to that of going from a typewriter to a computer. When I write my cursive, hand-written notes in Microsoft’s OneNote program, all the physical drawbacks of using paper disappear. If, for example, I want to fit a word on the last line, I simply shrink my previous writing, and write it down. If I want to move my words around, or shift my paragraph order, it’s even easier than it would be in a typing program. Searching through my notes is the best feature though; everything written down is recognized and quietly stored as text where it’s always searchable. These nice features are great, but they represent something bigger: digital paper.

Digital paper is a neat concept since it brings input from the physical world to the digital one: a tablet computer with the required hardware. For writing, however, a digital pen is best, whereas finger input is better suited for manipulating buttons and interfaces. A slate tablet computer, an analogue to a pen and a pad of paper, would therefore need to have an interface designed for both finger and pen use cooperatively.
Apple decided to leave out the pen, the full operating system, and the digital notebook application. Apple pretty much made the decision to make the iPad a big iPod touch, which is problematic. Earlier, Jobs talked about two different categories of computers: mobile and regular. A mobile device, like an iPhone or iPod touch, is best at receiving content. Watching a podcast, checking email, listening to music, and playing small games are tasks well suited for mobile devices. A computer is best for creating content: writing documents, typing emails, and creating movies, music, and games. In order to be middle ground, you have to be good at both. The iPad, being only a big iPod touch with a restricted OS, seems for many uses such as essay outlines and storyboards. This is how I want to write my essays; if I have the ideas in my head, why should I stress getting them written down correctly? Tablets will bring a revolution in computing and content creation; the way we get ideas onto paper won’t be so difficult if there’s no paper. It’s simple, just express the ideas however you can, and let the tablet do the work of piecing it all together.

While this seems like a futuristic dream, some aspects of it are being created present-day when designing tablets. The Microsoft Courier concept is a tablet device that merges some of these ideas. In the demo video, an artist writes in her “infinite journal” which works similarly to OneNote. Added is the ability to merge ideas from different sources, kind of like a digital organizer for your thoughts and ideas. This is surely a step in the right direction, one that will get us to the ubiquitous tablet.

The mere fact that people are discussing the iPad is important because it’s generating ideas about what tablets should do. Where exactly does a tablet device fit into the computing paradigm? A tablet has to be best at both creating and receiving content. As a digital replacement for paper, the tablet can excel greatly at both. While the iPad’s future success as a single product is uncertain, the effect it has on the industry is clear. I’m already forming iPad ready multitouch apps for my CS 3750 design class. So, if you’re using an iPad to read this article of The FIREWALL, or write one, you can be certain that you’re not going to be the only one with a tablet for very long.

“As a student, I want a tablet to be the only thing I bring to class, replacing all books and paper notes”
BY CHRIS RUSSELL

We’re smack in the middle of job application season at the moment, and many students are having to figure out how to apply for their first job that doesn’t involve flipping burgers. Career Services and the College of Computing advisors have an armory full of useful information on the subject, but, until you’ve gone through the process yourself, it’s difficult to really know what it’s like.

While a full-time position is still a few semesters off for me, my search for a summer internship is quickly drawing to a close.

COMPANY 1

For Company 1, the whole process started at a Tech Talk in mid-Fall 2009. One of Company 1’s software engineers was on campus to give a talk about one of their products and encourage students to apply for both full-time and intern positions. Their products really sparked my interest, so after he finished his talk, I introduced myself and handed him my resume.

Fast forward to winter break, and I got an email that said their recruiter liked my resume and wanted to set up a phone interview with me. I said I was still interested in the position, we set up a time for them to call, and I played the waiting game until the time came. The interview started out simply enough. They asked a couple of logistical questions (like what date could I start and whether or not I could receive a security clearance) and what my specific interests in the company were, then went over my resume for about five minutes. After that, the serious questions started. All said and done, I was asked three questions, all of which were heavily data structures and algorithms oriented. All three involved using a data structure to solve an unusual sorting or mapping problem, and all three were things I had never considered until forced to think about them. These took up about 2/3 of a 40 minute phone call, which left me a few minutes at the end to ask questions of my interviewer.
A week later, I got an email inviting me out to do an on-site interview at the company's main office. They offered to pay my airfare, room, and board for a two-day stay, and wanted to know if I was free two weekends from when I received the email. I said I was, they arranged my travel plans, and sent me my hotel, flight, and rental car information.

I flew into town on a Thursday afternoon, crashed in a (very nice) hotel for a night, and left for my interview early Friday morning. I followed their directions to their office, checked in, and was given a badge and a cup of coffee. After a short explanation of how the day was going to progress, I was ushered into a board room, and my interviews commenced.

Before the day was over, I talked to four different software engineers for about an hour apiece. Two spent a few minutes skimming through my resume, but the vast majority of each interview was spent working through problems similar to the ones I had in my phone interview. One interviewer didn’t mind me writing out my ideas in pseudocode, but the rest handed me a dry-erase marker and asked me to program my ideas on a whiteboard. One had me balance an AVL tree, another had me use a hash table to remove duplicates from a list, and another even had me turn a number into a string by converting it to base-26. All in all, I saw some really cool problems, but it was an exhausting couple of hours.

After the interviews ended, the engineers took us to lunch (on them, of course), and brought us back for a product demo. This was particularly cool, as it gave us a chance to see a lot of things they hadn’t shown us at the on-campus info session and also gave us a more hands-on view of their product. The day ended with a short conversation with the company’s head recruiter, who answered my logistical questions, and said I’d hear back from them within the week. I had the rest of the day to myself and took the chance to explore the city a bit.

After getting back, it took almost two weeks to hear from them, due in part to a rather amusing game of phone-tag that lasted for two days. Unfortunately, the company didn’t make me an offer, citing my lack of coding experience, but said to get back in touch with a year or two more of experience under my belt.

**COMPANY 2**

Company 2 was an Atlanta-based company that, thanks to a bit of blind luck, I stumbled across at the spring internship fair. Walking by their table, I noticed a position that seemed to match my skill set, and asked the recruiter a few questions about it. She was more than happy to answer my questions and just as happy to take my resume. I received a phone call later that afternoon encouraging me to sign up for an on-campus interview later that week but, unfortunately, it clashed with my interview with Company 1. She said that was no problem, and said she’d get in touch when they came back to campus later that month.

Sure enough, a few weeks later I received an email encouraging me to sign up for an interview time slot on Career Buzz. I picked one, emailed the recruiter to confirm the time, and dusted off my suit.

The day of the interview, I signed in at the Student Success Center and met my recruiter outside my assigned room. While the interview with Company 1 was highly technical, this interview was definitely a character-driven one. Rather than asking me to code complex algorithms, my interviewer walked me through my resume, asking me questions about problems I’d encountered, how I solved them, and what I learned from them. It’s important to note, however, that she did not ask me for generic fluff. In other words, she didn’t ask me to spout off a series of buzzwords and corporate rigamarole. She wanted concrete, specific examples from my classes, research, and
organizations about problems I had solved, and what I had done to solve them.

Afterwords, she gave me a chance to ask any questions I had, which I did, and handed me a thick application to fill out. I completed it, wished her a happy weekend, and sent her a thank you note later that night. She responded the next day and let me know I’d hear from them sometime in mid to late March about the rest of the selection process.

COMPANY 3
Company 3 was one I had been planning on applying to even before the internship fair. I made sure they were attending the fair, found their booth, and sought them out as the first company I spoke with. The recruiter was happy to talk with me, looked over my resume, and suggested a few groups within the company that would be a good fit for my skills. She said she’d submit the paperwork for me, gave me a handful of pamphlets, and invited me to an info session later in the week.

Two weeks later, the recruiter emailed me and asked if I was interested in an on-campus interview. I told her I was, she sent me a list of available times, and we got everything scheduled as quickly as possible.

The interview itself was definitely a technically heavy interview—preceeded, as always, by a brief scan of my resume—though it wasn’t quite as intense as my phone interview with Company 1. My interviewer only asked one fairly straightforward programming question, and he seemed much more interested in my thought process of getting to, testing, and optimizing my solution than he was in my solution itself. He left time for me to ask a few questions, and said I’d hear back in a week. I left feeling a little uncertain about my performance, and wasn’t expecting good news when the following week drew to a close. However, to my surprise, they invited me out to do an on-site interview at their main office.

Travel plans were made much as they were for Company 1—though, as Company 3 was significantly larger, the process seemed a great deal more organized—and I received a stack of prep material as tall as my desk. I reviewed as much as I could, approved the travel plan they sent me, and got ready for another intense weekend.

After a quick taxi from my hotel to Company 3’s main office, I checked in, grabbed a cup of coffee, and talked with a few other interviewees while I waited for my recruiter to get a chance to meet with me. She was incredibly helpful, explaining how the whole process was going to work, filling me in on what position and group I was being considered for; and even offering me some advice on how to do well. After a brief run-through of my interests and last-minute questions, she pointed me in the direction of my first interviewer and wished me luck.

As with Company 1, I spoke with four different interviewers throughout the day, though, unlike Company 1, I moved from office to office, while my interviewers stayed put. The interviews themselves were fairly similar to my on-campus one. They gave me a set relatively straightforward problems to code, and, again, seemed more interested in the how and the why of my thought process than in how perfect my final answer was. In addition to these, they also asked me a few abstract, open-ended questions like, “If I told you to do task X, how would you go about doing it?” or handing me a random object from around their office and asking how I would go about designing and testing it.

Each interview lasted between and hour and an hour and a half, meaning I started mid-morning and finished mid-afternoon. I wrapped up the day with my recruiter, who told me I’d hear within four or five days and gave me the address of her favorite local restaurant. I thanked her, checked out at the front desk, and went to grab some dinner. Halfway through dinner, however, my phone rang and, upon answering, was told that I was being made an offer for the summer. After an excited conversation and a few quick thank you notes to my interviewers, I booked it back to the airport and caught the red-eye back to Atlanta.
This year, the College of Computing and the School of Literature, Communication, and Culture have teamed up to add yet another country to the long list of places visited by their students. For years, computing students have been able to spend their summers in Oxford and Barcelona and any semester they choose in Metz, but this fall, some students will be going a bit farther north: Denmark.

The new program is set up as an exchange between Georgia Tech and the IT University of Copenhagen, a university specializing in digital design and new applications for digital media. Tech students interested in studying in the Scandinavian country can spend a semester or—if they so choose—an entire year studying at Copenhagen ITU. Shannon Dobranski, Associate Director of Undergraduate Studies for the School of Literature, Communication, and Culture, said, “The Copenhagen ITU program is a foreign exchange program that allows our students and their students to study at each other’s universities. It’s a good match for CM especially—but also CS—because it focuses on design.”

The coursework there gives students some background in design and some of the social consequences and contexts in which digital media is being used...It gives you a theoretical understanding of the current position of digital media and also some insight into how to best design digital media.”

The program is readily accessible to most students, regardless of their progress in their degree track. According to Dobranski, as the ITU program focuses more on the design and sociology of digital media, most computing students at Tech won’t encounter many technological barriers. She said, “There’s a lot less computation on the Copenhagen side of things... Our students won’t have any difficulty fitting into their program.”

She continued, “They’re going to give you a lot of design expertise...and they’re going to give you a lot of the sociology of digital media—sort of the effect of digital media on society.”
Of course, since the program is an exchange between the two universities, students from Copenhagen ITU will be attending Tech as well. Currently, the program’s directors anticipate hosting 3-4 students for the first round of the exchange and expect rapid growth once the program is well-established. The students who are coming to Tech will be in the second year of their bachelor’s program.

The first round of exchanges is set for Fall 2010, but the program has been in the works for a while. Dobranski said, “Our faculty and their faculty have been working on this for a number of years now, and [they] already have an established [relationship].”

Something that sets Copenhagen apart is the fact that it is, without a doubt, a full semester exchange program. The program lasts the full 16 weeks of the semester, and students can expect to return with the credits to show it. Dobranski said, “In general, the students who study in Copenhagen will return with 21 hours. That credit will be divided between CS and LCC classes, but we haven’t worked out which course numbers yet… [However,] the credit will transfer back in a way that will be meaningful for computational media and computer science students.”

For in-state students, all 21 hours will count towards hours attempted for HOPE. Additionally, as with many international programs, out-of-state students pay in-state tuition while studying abroad.

According to Dobranski, all said and done, the cost of studying in Copenhagen is comparable to the cost for an in-state student to study in Atlanta. While she does advise that the cost of living in Copenhagen is high, the economy is friendly towards students, transportation is cheap and high-quality, and students who choose to cook for themselves rather than eating out every meal can expect to save a large amount. She said, “By the time you pay for rent there, and modest dining, it comes to about what you could expect to pay here for dining and dorms.”

Tech students who take part in the program will come in during the fall or spring (or both) of the second year of Copenhagen ITU’s three year bachelor’s program. Their curriculum is a bit more

“THE PROGRAM IS READILY ACCESSIBLE TO MOST STUDENTS, REGARDLESS OF THEIR PROGRESS IN THEIR DEGREE TRACK.”
structured than Tech’s, meaning that their schedules are less flexible. A course breakdown is available as a sidebar. That said, there is room for flexibility in student’s curriculums. While the expectation is that most students will follow the curriculum described here, students who find a research project or another class (in English!) that piques their interest can take advantage of it. Dobranski said, “There [will] be a way to manage credit for other classes there. We design the program so that students are taking certain set classes, but you can deviate from that if you discover that there’s something else being offered in English, or if there are any research opportunities.”

Copenhagen ITU’s educational model is different from that of many universities in the United States. As Dobranski put it, “Work at ITU tends to be collaborative and self-paced...You’ll meet with faculty maybe only once a week for an extended lecture, and then you’re on your own or on your own with group members to build on what you learned...It’s a bit of different learning model, but it means that it’s a very collaborative setting.”

The applications for the fall program was originally due on Feb. 15th, but received an extension due to remaining uncertainties about some details. Students interested in the spring 2011 semester can apply until Sept. 15th. The early deadline is mostly due to the paperwork that has to go through. Dobranski says that while the visa application isn’t painful (it mostly involves mailing a passport to the Danish consulate and waiting to get it back), it takes time. This is in addition to the standard application to the Office of International Education and to Copenhagen ITU itself.

Anyone who thinks they might be interested should consider applying. Dobranski said, “This is a special opportunity for Georgia Tech students. It’s a great way to experience another culture, and especially to experience one of the unique cultural offerings of Denmark by studying design in Copenhagen, which is such a center for design and fashion and style.”
CPT (Curricular Practical Training) is the work authorization by which students can work up to 12 months in a program recognized by Georgia Tech. This is the kind of work authorization which one has to do before graduation. The other authorization is OPT (Optional Practical Training). Now, the traditional 12 months that the International students used to use is still there, but an amendment has been made: the work period for STEM Majors (Science, Technology, Engineering and Mathematics) has been increased by 17 months. This gives the international students a total of 12+12+17=41 months to work and complement their university education.

Everything until this point looks good, but sadly this is not the case. First of all, many employers handle data that is private or for national security and due to new measures by the Obama administration, are forced to hire only US citizens. A lot of good candidates are turned down by big companies like Lockheed Martin, FBI and Northrop Grumman because of their citizenship status, and although it may seem unfair, it is standard policy. Secondly, only Co-op jobs can be considered to be a part of CPT and since Co-op jobs do not attract large companies as much, the option really loses its charm. It effectively takes out 12 months out of the available 41 months to work. Thirdly, if Co-op exceeds even one day more than 12 months, the entire OPT is cancelled! The entire process is very delicate and must be well researched in advance.

For all international students out there reading this, be very careful in what you do so you don’t get defeated by the bureaucratic red tape.

BY MOINAK BANDYOPADHYAY

Let’s get started with a quick quiz:
What is the maximum amount of time that an International student can work in the US after graduation? And yes, this means legally. The choices are (A)Forever, (B) Until Chuck Norris approves of it, (C) 12 months, (D) 41 months, or (E) 15 years, 2 months and 25 days.

If you answered A, B or E and you are an International Student on F-1 or J-1 visa, this may be a huge wake-up call for you. According to the Rules and Regulations, an international student cannot work here forever without other necessary validation like a Green Card or an H1 Visa. If you chose B or E, you probably are a dawg from UGA and do not deserve to read this.

International Students not only have to worry about finding a job in the present economic crisis, but also have to think about their work authorization. There are two types of work authorizations that the students can primarily obtain – Curricular Practical Training (CPT) and Optional Practical Training (OPT).

Now going back to the question, the correct answers can be one of either C or D. A lot of students would think that once they graduate, they have only about 12 months of work left before they decide to get an H-1 Visa or get into graduate school; but that is in fact incorrect. The correct answer is D – 41 months. Let us discuss what the different breakouts are:

“ALOT OF GOOD CANDIDATES ARE TURNED DOWN BY BIG COMPANIES...”

The entire process is very delicate and must be well researched in advance. For all international students out there reading this, be very careful in what you do so you don’t get defeated by the bureaucratic red tape.
Ready, set, hack! This year's Hack Week ran from March 8th through March 12th, and provided Tech students with a unique opportunity to show off their skills. Hack Week, sponsored by Yahoo!, is one of the most exciting yearly events in the College of Computing. This year, Yahoo! Web experts led discussions on the latest news in computing, gave hacking tips, conducted coding workshops, and gave lectures about geolocation services, Hadoop, YQL, and YUI. In order to help with coding, multiple APIs and tools from the Yahoo! Developer Network were provided. The week concluded with the University Hack Day competition, where students exhibited their coding skills in a fun, yet challenging environment.

Students were given 24 hours to design some sort of hack, such as a web application or a widget of some sort. Participants were required to present their hack within a limited time frame of about a minute and a half by showing how the hack solves a respective problem. It was strongly advised that the competitors first state the problem and then the resolution, leaving the “how it works” part for last. Web Applications were typically presented and displayed on Yahoo!'s computers, but presenters could present widgets and other software on their own laptops. Coding began on Friday at 1pm and ran until Saturday, when the products, completed or not, were presented.

This year, Tech had over forty entries in the hacking competition. First place prize went to Shauvik Choudhary, Sahil Miglani, and Utkarsh Shrivastava for their hack, “Video in Ymail”, a Firefox extension that allows you to record and send videos through your Yahoo! mail. Second place winner was Roger Pincombe’s “Importance Filter for Gmail”, which analyses the contents of your inbox and sets apart those you are most likely to reply to or star. Mike Hirth and Daniel Hopper got third place for their “Lyricist” hack.
The Minorities at the College of Computing (M@CC) held its annual Throwback event on Tuesday, February 16th. Each year, Throwback is themed to give a certain era or genre of gaming -- board games and video games -- the spotlight once again. Previous years have highlighted 80’s-90’s videogames and last year’s event was centered on multicultural board and video games from around the world (Throwback International). This year, M@CC focused on music gaming with games like Guitar Hero, DJ Hero, and Rock Band -- inspiring the event’s title, Throwback Hero.

Throwback Hero was arranged as a pick-up-and-play arcade, with TVs lining both sides of Klaus 1116 and consoles occupying both projectors. Thanks to many student volunteers from around Georgia Tech, a large spectrum of consoles was represented. There were a few Xbox360s, a few Playstation 3s, a Wii, and a Playstation 2, all of which offered students a number of platforms for game play. The list of games included Guitar Hero 5, Guitar Hero 3, Rock Band 1 and 2, and DJ Hero. Groups of students typically played in teams of four on the projectors in Klaus tackling Rock Band 2’s song list with one particular student exploring his vocal skills with Blink 182 -- many, many times. Students enjoyed being able to play with similarly-skilled teammates, as evidenced by the moments of synchronous head-bobbing and finger-twitching every student displayed during Expert play.

Attendees enjoyed the variety of games, pizzas, drinks, and snacks available at Throwback Hero, with many of the refreshments disappearing within the final fifteen minutes of the event. A number of students reported having played games they had never had the chance to play before Throwback Hero, and they appreciated the relatively short wait times to play all of the games.

As advertised, students had the opportunity to win prizes at Throwback Hero by participating in the “Throwback Score Attack: Guitar Hero 3” challenge. Contestants could sign-up in-person...
during the event or online, using the event website -- ThrowbackHero.com. Students were given one chance to play pre-selected songs on Medium, Hard, or Expert difficulty competing for the highest score amongst the contestants. Each difficulty level corresponded to a particular prize, rewarding the higher skilled players with better prizes. Although many students participated in the score attack challenge only three skilled players walked home with new speakers or headphones.

M@CC hopes to continue to provide its annual Throwback events to the College of Computing and Georgia Tech population, and is always looking for ideas for events. If you have any ideas for Throwback, events in general, or want to be more involved in the College of Computing community, feel free to join M@CC on Tuesdays at 6pm on the third floor of the College of Computing, Room 354. For more information, email terris@gatech.edu.
BY CHAITANYA ADGAONKAR

The Georgia Tech Programming Team is a group of about fifteen to twenty students who share similar interests, namely in algorithmic programming competitions. They primarily train to compete in the ACM International Collegiate Programming Contest (ICPC) South East US Regional every fall semester. The ACM ICPC is a multi-tier, team-based programming competition operating under the auspices of ACM. Tens of thousands of the finest students and faculty in computing disciplines in almost 2,000 universities from over 80 countries gather together to participate in this event. Quite simply, it is the oldest, largest, and most prestigious programming contest in the world.

Although the team competes in various competitions throughout the year, their most important competition is the ACM ICPC South East US Regional. Each team of three contestants, has one computer, and has somewhere between 8 to 10 algorithmic problems to solve in 5 hours. If the team wins the regional competition, it is invited to the World Finals, where teams from all over the world come to compete. In the past decade, the Georgia Tech Programming Team has been to the World Finals 6 times and has performed consistently well in the regional competition.

The team also participates in a variety of online competitions. These are usually held on various websites such as TopCoder, are typically individual competitions and last a little more than an hour.

"[Being part of the team] helps your skills in logically breaking down problems into sub problems to solve these sub problems," said Toprag Gurung, a graduate student who coaches the team. "These problems require a lot of creativity in solutions, so it helps you rediscover your creativity." While the team is coding away and having fun, the experience obviously greatly helps them in real-life situations. "It requires an extensive knowledge of algorithms so helps build your algorithmic knowledge. To be a part of the team requires you to be a good coder, [making] you a better programmer... it is also
a lot of fun," Andrew Ash says. Speed, efficiency, concentration and accuracy are some of the skills that can be honed as a part of the team he added. The team has received a lot of support from the faculty. Dean Foley, Cedric Stallworth, Meredith Goodman, Elizabeth Collums, Jennifer Whitlow and the College of Computing have been extremely supportive of the programming team. The faculty has provided the team with space, computers, food, transportation and most importantly, encouragement. The team is extremely grateful for their support. The team is very grateful to David Van Brackle, the former coach of the team at Georgia Tech, and now the Chief Judge at the South East Regional Competition. He has helped the Georgia Tech Programming team since 2001.

The team practices for the competitions throughout the year. The team holds weekly discussions on Thursdays from 6pm to 8:30pm on the CoC 3rd floor. Each week they have 3-4 problems that they work on. These problems are algorithmic in nature, and cover topics and ideas such as Graph Theory, Computational Geometry, Dynamic Programming, Greedy algorithms, Number Theory, and Data Structures. Gurung says "We encourage using Java... our philosophy is that programming languages are tools that we utilize to solve problems. Java is extremely programmer friendly, and also provides libraries... after all, it is not the language that matters, but the ideas that matter. In our case, the idea is the solution strategy and whichever language enables us to solve the problem, which over the years we have found Java to be extremely helpful in." On Sundays, from noon to 7pm, the team organizes simulated contests at the College of Computing, where they run a contest from 1pm to 6pm. From noon to 1pm, they discuss various ideas, and from 6pm to 7pm, they discuss solution strategies for problems in that contest.

The team has been performing well, and is looking to improve even further. Topraj says, “Georgia Tech has a top Computer Science program... We hope to welcome brilliant minds into the GT programming team so that the team environment can be [an] even more intellectually stimulating experience.” In March, the team will travel to compete in Mercer. This will be a warm-up practice for the regional competition in the Fall. Here is wishing the team best of luck from everyone at THE FIREWALL.
UPCOMING EVENTS
GET READY FOR AN EXCITING MONTH OF ACTIVITIES

COCAESAR’S PALACE
MARCH 30
CoCaesar’s Palace is hosted every spring by Women in the College of Computing (W@CC). This year’s theme is Film Noir, inspired by mystery movies from the 1950s. You can find Poker, Texas Hold ’em, Blackjack, and Roulette tables arranged around the Klaus atrium, being dealt by faculty and staff. Other organizations from the College of Computing participate by simultaneously holding thematic games and activities of their own. CoCaesar’s had a turnout of over 200 students in 2009, making it one of the biggest events from a student organization at Georgia Tech. Mark your calendars for March 30th, 2010.

NEW COC WEBSITE
MARCH 2010
The entire College of Computing website received a complete revamp, designed around input received from the College of Computing community. Aside from differences in layout and color schemes, the new website is more dynamic and organized, with a focus on the people that make up the college and other computing professionals. You can judge how the new website reflects the success of the College of Computing today. Have a look by visiting www.cc.gatech.edu.

PROGRAMMING COMPETITION
APRIL 16
Think you can code? The GT Programming Team is organizing a coding competition where you can see how you stack up! On April 16th, starting 6:00 PM, you can expect to see problems over a wide range of difficulties in an evening complete with food, fun, prizes, and balloons. We hope to see you there!