Attitudes about Computing in Postsecondary Graduates

Michael Hewner
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, Georgia
hewner@gatech.edu

Mark Guzdial
Georgia Institute of Technology
801 Atlantic Drive
Atlanta, Georgia
guzdial@cc.gatech.edu

ABSTRACT

Computing educators may hope that postsecondary courses both convey content and also give students a new perspective on computing. In the study described in this paper, a sample of students about to graduate with their postsecondary degrees wrote about their relationship with computing and what influenced that relationship. Computing majors wrote expressively about the excitement and breadth of the discipline. Other majors were positive about computing, but the essays indicate that postsecondary education (including introductory computer science courses) did not have a large effect on their attitudes about computing.

Categories and Subject Descriptors

K.3 [**Computers and Education**]: Computer and Information Sciences Education

General Terms

Experimentation, Human Factors

Keywords

CS1, programming, non-majors

1. INTRODUCTION

We teach non-major computer science to convey the big ideas of the discipline. Educators have stressed that computer science concepts is important in grasping how computing impacts everyday life [17] and that the principles of computational thinking can be applied elsewhere [20]. We also hope students will learn the practice of programming and find it to be a useful skill in their later careers.

Beyond these conceptual goals is the potential to change student attitudes. Computer Science is a field that is frequently misunderstood by students. It is widely believed that a well-designed introductory course has the potential to make students more excited about computing. The building of more accessible introductory courses has frequently been suggested as a way in which we can improve the image of computing and potentially attract new students. For example, in the article “Revitalizing Computing Science Education” Mahmoud lists fun introductory programming courses as one of his five recommendations for improving CS education [10]:

> Make CS courses fun. Design computer programming courses suitable for students in the arts. This last measure–programming for poets–might eventually motivate such students to move into CS, or at least minor in the subject.

Mahmoud goes on to say, however, that the “core solution” to CS enrollment problems requires intervention before college. Intuition suggests that better introductory courses might be useful, but maybe not enough to change students’ attitudes and solve enrollment problems.

It has been shown that college as a whole has the potential to change student attitudes in a variety of areas [12]. These gains are usually modest but can be significant. Less positive results have historically been found with regard to changing majors. Students often change majors in college but unfortunately much of this changing can be attributed to switching between related majors and a general trend of leaving science and engineering majors[1]. Still, given that some attitudes do change in college it is worthwhile to ask what effect college has on students attitudes about computing and what courses in computer science can do to improve these attitudes.

All students at Georgia Tech are required to take a mandatory Computer Science course with a heavy emphasis on introductory programming. Three different courses are offered: a Python or Scheme course for Computer Science majors, a MATLAB-based course for Engineering students, and a Python-based Media Computation course for other majors that focuses on manipulation of pictures and sounds. All of these courses have been designed to present real Computer Science concepts in a way that is relevant to the students’ interests.

These introductory Georgia Tech courses were designed to improve poor student success in introductory CS. They were also intended to address widespread student opinion that the traditional introductory Computer Science course they were required to take was not relevant to them. Previous studies have shown that the more contextualized approach resulted in better student performance and more positive opinions about the course and CS in end of course evaluations [8]. There was also evidence that some students used the pro-

ICER’08, September 6–7, 2008, Sydney, Australia.
Copyright 2008 ACM 978-1-60558-216-0/08/09 ...$5.00.

gramming they learned in introductory CS later in college [7]. Students reported that they felt that what they learned was inauthentic in terms of degree or career, but useful for their lives. Even though students enjoyed the course the majority were not interested in taking more Computer Science courses [19]. This suggests that students are attitudes about computing are more than a simple range of liking or disliking computers, and that some aspects of student attitudes are easier to change than others.

Over four years since the introduction of these specialized courses, we now have a unique opportunity. By most of the usual measures, these courses are good. They have been designed to motivate students and to present material in a way that draws on their existing interest. Do students talk about these courses as having a significant impact on their relationship with computing? If not, what factors are drawn out as important in their experiences with computing?

A good deal is known about what student attitudes about computing are before they enter college, but less is known about how the college experience affects these attitudes. In college students are exposed to how computation is used in their chosen discipline, and get plenty of opportunities to broaden and improve their computer skills. All these things have the potential to change student attitudes.

This paper describes a autobiographical study undertaken at Georgia Tech to uncover attitudes about computing among undergraduate seniors. We begin by discussing existing research concerning perceptions of computation. Then we discuss the details of the autobiography study and the analysis. We discuss themes that emerged from the essays, illustrated with examples from the autobiography text. Finally we conclude with some potential implications of the themes we discovered.

2. RELATED WORK

Psychology research has investigated attitudes from a wide variety of perspectives, but there is no universally recognised definition of what an attitude is. An appropriate model of attitudes for this research comes from Rokeach [14] who describes an attitude as a "relatively enduring organization of beliefs about an an object or situation predisposing one to respond in a preferential matter." The attitudes that Rokeach's theory was used to describe were social issues like race integration and politics – attitudes of a similar sort to the questions we asked about computing.

In Rokeach's theory of attitudes, some beliefs are central and some beliefs are peripheral to the overall attitude. Central beliefs are expected to be the most resistant to change, and also have the greatest effect when they do change. But what makes this theory the most interesting for our study of computation is the evidence that what distinguishes individual attitudes of groups who disagree (e.g. liberals and conservatives) is not so much the content of the individual beliefs but what beliefs are central[9].

What Rokeach's definition suggests about computing attitudes is that attitudes about computing are not a single issue of liking computers or not. Two people may agree that computers are both important as communication tools and vehicles for exploring mathematical ideas. But person who considers mathematical exploration to be the central use of computers is obviously going to have a different approach to learning and using computers than a person interested in communication. What aspects of computers students feel are central can have a major effect on their actions, even if their beliefs are similar to other students.

In CS education research understanding how students view their relationship to computers is a problem that has been approached from a variety of angles. Declining CS enrollment and high student attrition even after they have entered the major [4] has driven significant exploration into what factors influence students to pursue a major in Computer Science and what factors are significant indicators of success in their introductory Computer Science coursework. Carter [5] used surveys to determine that high school students generally had inaccurate (or no) perceptions of CS as a field prior to college. There have been a variety of studies that attempt to find factors that predict CS1 performance [6][2]. In one of the few studies of students near graduation, Biggers surveyed graduating seniors and discovered several significant differences between graduating CS majors and those who leave the major prematurely, including a focus on Computer Science as more than just programming in those who stayed in the major [3].

To understand complicated issues like student attitudes, though, qualitative studies can offer a more complete picture. Margolis and Fischer [11] present a deep analysis of gender issues in the CS program at Carnegie Mellon. One of the findings of this work was a stark difference between men and women's CS majors' attitudes towards computing; men were more interested in techonology for its own sake, while women were more interested in the potential applications of technology. The media-computation oriented Computer Science courses at Georgia Tech have been evaluated by qualitative interviews immediately after the course conclusion and by brief surveys one year after the course end [13]. This study indicated that the more contextualized CS teaching approach improved attitudes about programming.

Carsten Schulte and Maria Knobelsdorf [16] studied attitudes concerning computers by asking students to write computing "autobiographies". They compared the autobiographies of senior-level psychology students with incoming freshman Computer Science students. Their results show a large difference in the way the two groups talked about their experiences with computing. Schulte and Knobelsdorf characterized the autobiographies of the psychology majors as coming from an outsider perspective, and being focused primarily on the use of computing to accomplish specific tasks. The psychology majors frequently referred to having to ask for computer help from others. The computing freshman considered themselves insiders, and thought of their activities as oriented towards designing rather than simply using. Their descriptions of interactions with computers was characterized by "playful exploration" in a way that facilitated their transition from initial use to thinking of computing as designing.

We were interested in using Schulte and Knobelsdorf's research method but with several important differences. We wanted to study students across a wide variety of disciplines, to get students with varied college computing experiences. We were also specifically interested in computing majors, to see how their relationship with computing differed. For all of them, we were interested in their attitudes toward computing and computer science.

We choose to focus on seniors because they have the full college experience to shape their attitudes. We also felt it was likely that seniors had a specific perception of how computing was used in their future careers. The student body at Georgia Tech was also likely to be technically oriented than average, simply because they elected to attend a technical school, which made the student body even more likely to have strong and interesting attitudes about computing.

3. METHOD

Participants were asked to write a short autobiography about their experiences with computing with several short sample excerpts to guide their writing. Autobiographies have several advantages: they are freeform, with less chance of an interviewer implicitly influencing the participant with his questioning. The narrative structure encourages the participant to reflect on experiences, and to come up with an approximately coherent story that reflects their attitudes and highlights the most meaningful experiences. Finally, the autobiographical format is logistically easier to administer, making it possible to get input from a greater range of participants than with interview style qualitative analysis.

We recruited Georgia Tech students for the study who identified themselves as being approximately one year away from graduation. They are chosen from three groups, based on the college of their chosen major – College of Computing (Computer Science and Computational Media), Colleges of Engineering and Science (Chemical Engineering, Mechanical Engineering, Psychology, Biology, Mathematics, etc.), and other (including Public Policy, Architecture, Management, etc.). These groups were chosen to as very rough separation between students who likely did significant programming as part of their major (Computing), students who might have used programming later in their curriculum (engineering and the sciences), and majors with little direct need for programming (other). All students chose to attend a technical school, so this is not an accurate cross-section of the population as a whole. But it is reasonable to say that these students represent a variety of approaches and uses of technology. All of them had at least one introductory CS course that focused on programming.

We recruited students in a variety of ways. Fliers were posted in buildings all around campus, both in the campus public areas as well as in buildings specific to majors of the various college groups. We also presented fliers to students in classes of various majors (psychology, computer science, management). Everyone who participated received a US$10 Starbucks gift card as compensation for their time.

Each participant was directed to a webpage. After getting information about their college and agreeing to a consent form, they were taken to the following question:

> We would like you to write an "autobiography" about your experiences with computers. You might want to write about a certain class, a computer game or more in general about your thoughts about computer usage and computer science. How do you use computers now? How did you first get introduced to computers? What people or experiences had the greatest impact on the way you use computers today?
>
> Use the four short excerpts below to help inspire you.

The excerpts themselves were chosen carefully. Some of them were from respondents to Schulte and Knobelsdorf's study (translated from German) while some were fabricated. Two of the four were positive about experiences with computing, and two were negative. Three out of the four focused on college experiences rather than early life ones. Three out of the four mentioned programming, and three of the four also mentioned non-programming computer applications. The purpose of the samples were to influence slightly the autobiography writers while still allowing them to focus on the experiences they saw as most significant. We wanted them both to write about the later college experiences that we were particularly interested in as well as focusing on experiences rather than lists of technologies. Following the main autobiographical question were a small number of specific questions with regards to gender, which introductory computer science course they took, and how they heard about the study.

2	programming is valuable for my job
3	programming is fun
4	using programming in classwork after CS course
4	personal programming projects
4	bad preconception about programming
5	my peers are better at programming than me
9	programming is not relevant to what I do
11	writing HTML

Table 1: Example programming-related codes and frequencies (across essays)

The results of our recruitment broke down by college as follows: 12 engineering/science students, 8 computing, 7 other. Eleven of the respondents were male, 16 female – which was surprising considering the overall undergraduate population is only 30% female. Given the good female response rate, all of the themes described below are evidenced in essays from both male and female participants.

3.1 Analysis

The 27 essays were analyzed with a grounded theory approach of open coding followed by axial coding as described by Strauss and Corbin [18]. Each sentence (and occasionally larger chunks) were coded. These codes were then combined into more general axial codes. These axial codes were analyzed for prevalence across of all the essays and cross-referenced with each other to discover any distinct groups. These frequently occurring axial codes formed the basis for the themes in the following sections. Particularly detailed coding attention was given to attitudes concerning programming and Computer Science courses – see Table 1 for some example programming codes and their frequencies.

Quotes in this paper are reproduced verbatim from the essays, including grammatical and spelling errors.

3.2 Potential Risks to Validity

The biggest risk to validity with the autobiographical format is that the essays only record what the students choose to write about as significant. Bad experiences can get recast in a positive light or omitted altogether. Furthermore, there is no guarantee that students know what was actually significant – these narratives record what is perceived as significant, and therefore more about the students' current perspective than an accurate historical account. But for exploring self-perception, the autobiography provides a deeper view than might be obtained from more direct questioning.

Another problem is because of the open-ended nature of the autobiography question, we can't look at what students omitted as a source of viable data. If it was somewhat surprising computing students did not mention specific classes or computing projects, we cannot assume that they weren't involved in such things; perhaps they simply opted to focus on the broad view of the field and excitement because they felt that was more relevant. We have to draw our conclusions from what they did say.

Finally, even though careful effort was made to acquire a good mix of Georgia Tech students, it is likely that students who were more excited about computing (or more willing to write essays) responded more frequently that others. It is clear that this sort of study is in no way intended to be statistically representative – the themes that these autobiographies identified hopefully exist in similar students elsewhere but the fact that they occurred frequently in these autobiographies does not suggest a similar frequency elsewhere.

4. THEMES IN THE ESSAYS

4.1 Computing Majors' Narratives

The narratives written by College of Computing seniors have a very different feel to them then those of non-majors. In our coding we did not separate the various majors but once axial codes began to emerge it was clear that the categories for non-computing majors and computing majors were distinct. This is not too surprising: computing majors are discussing a topic that's obviously been central to their lives for the past four years, and one they had some reason to choose as a major. This interest and the autobiography form's implicit encouragement to create a coherent narrative means that we expected some interesting stories.

4.1.1 Coming to Computers

The narratives do not speak to what experiences drew them to computers initially. Computing majors talk about always being interested in and fascinated by computers. The stories can include epiphany experiences in which one pivotal moment is mentioned. This is an experience that makes the writer aware that their existing strong interest in computing can be turned into a career, not a good experience with computers that made them like computers:

> "Quite casually, [my English teacher] mentioned that I seemed to love computers, and that I might enjoy a job using them. It was as if suddenly I could see in color–I'd NEVER thought that computers could be a job."

Sometimes these stories are quite exciting, talking about the writer overcoming various obstacles or seeking out computing opportunities. Students mention pre-college programming experience and occasionally personal projects. But these specific activities appeared with similar frequency in the essays of non-computing majors. Often a Computer major's story can be quite engaging without mentioning specific computing activities:

> "My journey has been full of excitements. Since the beginning I have enjoyed computers. I did my schooling in India. After 12th grade I wanted to go in Computer field but computers were new at that time...so I went for Mechanical Engineering. But my liking for computers never died. I started some private computer courses on the side. I came to the US in 2000. After coming here I started part time job in a store..."

In the stories of computing majors, the narratives often end once the student enters college. This is despite the fact that we explicitly choose examples to focus on college experiences, in order to (implicitly) encourage students to talk about college.

> "I read the guide to my TI-89 calculator a dozen times, learning to make simple games or programs in it. I was beside myself with joy. To me, computers are freedom, they are entertainment and above all they are a symbol of power and adulthood."

4.1.2 Computing: Expansive and Exciting

For the computing majors who do talk about their college experiences, coming to college does seem to be coupled with a significant change in attitude which they often associate with their courses (rather than friends, faculty outside class etc.). College portions of the essays often talk about being surprised by the breadth and excitement of computing as a major. Usually, they do not mention a particular subfield or technology that is exciting to them. It seems that for computing majors, the full academic expanse of the computing field is exciting to them:

> "Walking into GT and taking my first CS class taught me that computing is a field more expansive than I had even imagined....I learned a lot about the vast amounts of research, work, innovative thought and hard cold sweat that had gone into developing the world and art of computing. Now that everyone needs the skills of a computer specialist, I know that I will play an important role no matter where I work."

> "To be honest I do love it! Even though I do not know exactly what I want to do, well of course a software engineer, I still love it. And I loved the java course here, its my fav language so far. I still need to work on broaden my perspective by learning more diverse and different languages."

> "While at GT, I realized that the field of computing was much wider than I thought....I took CS classes whenever possible and as early as I could having met the pre-reqs...I still enjoy the technical side of computing but I prefer to bridge the communications and usability gap between information technology firms, professionals, and products and the people who desire to use them to their fullest potential."

Essays like this are evidence that the overall message that computer science is an exciting and wide field is being transmitted to (at least some) computing majors. Based on the way they talk about computer science, it seems that this change is more than simply learning new knowledge – it is a real shift in attitudes.

4.1.3 Negative Images about Computing

The narratives of non-computing majors almost never mentioned that computing was asocial or gave other negative impressions of the field as a whole. Non-computing majors' essays were not entirely positive – but they tended to limit their complaints to individual professors or technologies. Computing majors, on the other hand, frequently made references to classic computing stereotypes or problems, even in the context of their (very positive on the whole) essays:

> "My parents discourage the computer field due to the trends of outsourcing. I seemed to lose touch, but I still bought books and wrote programs instead of taking courses in high school."

> "After the end of my 2nd year, I recognized that while I still loved the fact that there were so many options within CS, I was beginning to distance myself from people by becoming so closely involved with technology and unique expertise. To be frank I was a little afraid of being sucked into the CS major stereotype of being a pale, scruff poorly dressed student who knew little more than gaming, hacking, and which hardware on the market was the best."

> "As a computer science major, I think I balance my life pretty well. I do not play computer games for hours on end. Instead, I enjoy skydiving, racquetball, and cooking. My interest in computers extends to how to manipulate them, but stops short of major recreation."

In general, these students seemed to feel that these stereotypes were something they could overcome (though there are also a few essays of students who left the major citing similar issues). But the fact that these issues were mentioned so frequently only in the essays of computing majors suggests that even when these problems are not really mentioned by Georgia Tech students as a whole, those in the computing field are much more aware of these stereotypes. Beyond that, they are even taking actions (not taking CS classes in high school, getting management certificates, etc.) to compensate for them.

4.2 Non-CS Majors' Narratives about Computing

4.2.1 Tech-savvy

The most obvious characteristic in the non-CS essays was that they were extremely positive about computing and their ability to manipulate it. Essays mentioned using computers frequently for schoolwork as well as for recreation – Facebook, games, email. With one exception, the essays did not mention frequently needing help for doing computing tasks.

> "In some of the organizations I have been in, I have had to create flyers, brochures, reports, emails, PowerPoint presentations...I now use the computer to do most of my shopping, whether it is for clothes, books, or electronics. I also keep track of my financial information using the computer. I'm sure I am not naming all the things I use a computer for, but I honestly do not know where I would be if I wasn't able to use a computer!"

> "I use the computer everyday and all day especially as a Tech student. I email people all day. I watch my favorite music videos all day on the computer. I am on facebook and chating with friends online frequently as well. I apply for jobs and find information I need to do labs, homework,and prepare for tests online as well. All in all, I could not imagine my life without it."

There are several factors that might make the autobiographies we received not representative. For one thing, students disliking technology are presumably not as likely to attend a technical school like Georgia Tech . Also, even though we made an effort to recruit respondents across a variety of majors and provided compensation, those excited about computing are more likely to go through the effort of writing a voluntary computer-oriented autobiography. Still, we think the respondents are clearly not an anomaly – nontechnical user definitely still exist (at Georgia Tech and elsewhere) – but the population of confident, technically experienced users that were prevalent in this autobiography study are population we need to speak to.

The technical experience of these users encompassed more than a familiarity with productivity applications. In our autobiographies we found that these users often had high school experiences in web design and a variety of other creative computing applications:

> "When I was in high school I took two animation classes, one on Flash and the other on 3D Studio Max. I also took a class in webpage design...I learned a lot from the class on Flash and the webpage design class, but the 3D Studio Max class was pretty useless. For our end of semester project in the Flash class, I made a movie that I still think is kind of impressive for what little training I had."

> "All of this work made me really excited to showcase my new talents. I made a video for my grandparents for Christmas with family pictures put to music. In terms of other, more advanced experiences I have had with computers, I took a computer science class in high school. We used C++ to learn the basics of computer programming."

Given the breadth of technical experiences our respondents described, it is interesting to speculate why they didn't choose Computer Science as a major. The kinds of computing experiences that the non-majors describe they are similar to the computing majors. The way computing majors describe the experiences are markedly different. Computing majors' tendencies to put their pre-college experiences in the context of a narrative about "finding" computing make it difficult to be sure if computing majors have always had a different attitude than non-majors; this way of talking could be something that has developed in college and changed the way they write about their past experiences.

What is clear is that when we think about changing the attitudes of students in introductory computer science courses, it is not an issue of making them like computers. From these essays, its clear there is a large population that already has a confident positive attitude about computing from even before college.

4.2.2 Tech Hobbyist

Just because students have a confident positive attitude, it would be wrong to think that the tech-savvy users we described above are stereotypical nerd users. There is another category of autobiographies that evidence an attitude which we referred to as technology hobbyists:

> "With few friends with the same interests and parents who had no idea computers could be 'hand built' I was on my own....After many of my friends said they liked my work, I created a photoblog. I was able to use many of the things I had learned from the past as well as new things that had became popular such as XML and Perl."

> "After several years of maintaining a fairly popular website, we got bored and began to make our own personal web pages for fun. I took a beginners programming class in high school and learned some visual basic and a little bit of C. I even wrote some very simple games. College was where blogging began to be all the rage. I got real into learning javascript and CSS."

Hobbyist mentioned learning particular technologies as fun. They were distinctive from the computing majors in that they wrote concretely about particular projects and technologies and did not speak so expansively about the field as a whole. They were distinctive from other non-majors in that they spoke about learning technologies outside of structured classes for fun much more frequently. We think these hobbyists are evidence of a component belief of computing as something fun to learn that is part of the overall attitude about computing. This belief comes up in other essays and is most prevalent in this particular group of essays.

A few of the hobbyist-style essays mention considering computing as a potential major.

> "As a freshmen in High School I was sure I would go to a school for CS and start a e-commerce web site. However, after the 2000 crash and the trend of outsourcing, I soon became disheartened."

Another essay mentioned dropping out of Computer Science because they found it too theoretical:

> "I really didn't like any of the cs classes I took in college. They taught me a useless fake program called Dr. Scheme. All I really learned was cs theory. I struggled in my cs classes, but luckily, I switched majors after freshman year into Architecture."

It is not too surprising that students in the hobbyist group consider computing a potential career path, but it is interesting that the hobbyist-style essay is generally different than those of computing majors. Perhaps this is because taking computing courses gives one-time hobbyists a more expansive view of computing as whole. Or perhaps hobbyist-oriented computer users are not being attracted to the major, despite their recreational interest in computing.

4.2.3 Non-Computing Majors and Programming

We have discussed three computing attitudes evidenced in our essays: computing as a broad conceptual field, computing as a source of productivity and confidence, and computing as a source of recreation. How does programming and introductory Computer Science fit into this picture?

The answer seems to be that liking programming may just be a part of a person's overall attitude towards computing. Rather, each of the three attitudes we saw tended to view programming in its own particular context. Each group felt that programming was useful, but only in the context of the central beliefs that formed the basis of their attitude towards computing.

We wanted students to talk about Computer Science and we encouraged them to do so when we posed the autobiography question. Most of the essays brought up programming at some point, and many mentioned their introductory course. Commenting on the experience of programming, students had variety of things to say:

> "In my first computer science class, I learned how to create a website page. I even learned how to code in SQL which was fascinating to me because I never thought that I would take interest in anything that was remotely related to computer programming!"

> "Programming to me is like a different mindset that I just didn't understand very well. I had multiple people try to help me including TAs and I studied a lot, but for some reason it never really clicked for me...I hate programing and all the technical stuff that goes on behind the scenes"

> "Taking this CS class has gotten me over my fear of computer coding, but just turned it into sheer annoyance."

Non-majors frequently described learning programming as a challenging experience: some decided in the end that they liked it, some decided that they didn't. Even for those who had programmed in high school, college-level programming was a new experience and one they felt was worth commenting on in their autobiographies (although, to be fair, the phrasing of the question was designed to encourage them to think about programming experiences).

Even when students said they liked programming, however, they still tended to think about it in the context of what they were already doing. The attitudes about computing stayed the same, and so complaints about relevance were frequent. Technically-savvy students tended to feel that the course would need to be related to their major to be useful to them:

> "However, I still thorougly enjoy helping my fellow peers who are taking CS1301 with their homework or any other material in that class... Unfortunately, because my major has no use for it, I have not programmed using Python since the CS1315 class."

> "In addition, only psychology majors interested in doing research would ever need to design an experiment using programming, so that means

> that not all of us would benefit from CS. Overall, the CS class I took was useless to me personally, and it did not make me want to take any other CS classes."

Students writing in the hobbyist style sometimes mentioned using programming in their spare time for personal projects. But the introductory Georgia Tech programming course was usually mentioned peripherally – its content was not relevant to the interests of these students either:

> "I was looking for a class that could teach about diagnosing common computer errors and the functions of each part within the system. However it seemed like just a novel way to write script using a programming system that I have not touched since. I instead learned about computer systems through CS majors that I made friends with."

> "I really enjoy the application software like Office and the some unique programs like WinDirStat, FTP, VOIP, IM software. I use the computer extensively for entertainment, research, productivity tasks, but have still not overcome my fear of programming. That is why I am yet to develop a web page that I would feel happy about."

Even though the introductory CS classes were built to appeal to students and place programming in a context relevant to students, essential attitudes about computers seem to be constant. Even if students learn programming or like programming, they seem to expect programming to be useful to them as part of their existing computing activities. This is not necessarily a sign that the Georgia Tech courses are bad – for that sort of assessment, the existing data about student success and enjoyment is likely more accurate. This is a sign that successfully teaching students the skill of programming and changing their attitudes to computing in a way they will bring up several years later are two different things.

5. DISCUSSION

It's usual to think of a positive attitude about computing as a single thing, but there seem to be distinct beliefs that are part of student positive attitudes:

- Computing as a variety of fun and useful applications
- Computing as interesting technology it can be fun to learn about
- Computing as a intellectually interesting and broad field of study

Almost all of our respondents said they like computing but which of these aspects is most central varies between students. Even when students have a positive experience in introductory computer science, frequently they nonetheless continue to think about computing in the same way. To place this in the context of Rokeach and Kerlinger's [15] ideas about belief centrality, the students' previous beliefs about computing may remain central while the things they learned in introductory computing change their peripheral beliefs in smaller ways.

It is reasonable to ask what attitude change is possible for non-majors. Though the themes uncovered in this study are not claimed to be in any way an exhaustive list of potential ways to approach computing, but it may be reasonable to say that enjoyment of programming as its own belief and tends to be viewed in the context of existing central beliefs about computation. Students at Georgia Tech have a large amount of experience with technology before they arrive: there does not seem to be a need for a course to alleviate anxiety about computers or make students like to use computer. Even though students were confident in their computer skills and used computers recreationally, most of the students who mentioned the enjoyed programming nonetheless dismissed it as not applicable to their goals. This is not unreasonable; as long as the overall attitude is still to improve day to day computing or enjoy oneself playing with technology, there are usually going to be more efficient ways to fulfill those goals than exploring computer science.

On the other hand, computing majors talk about their relationship with computers, they tend to speak less about the technical aspects of the experience, and more about their feelings about the field of computing as exciting and expansive. Their pre-college experiences are similar to non-majors, but those computing majors who talk about their experiences after high school refer to these earlier experiences as leading up to their present understanding. Computing majors often mention the fact that they were surprised by how much broader computer science is versus their expectations.

Considering the computing majors' essays through Rokeach's formulation of attitudes, we cannot be sure that real attitude change has occurred [14]. Asking a computing major about their relationship with computers places them in a different situation than non-majors: the computing majors may well feel under some obligation to 'sell' themselves or Computer Science in their essays. However the consistency of the computing majors' essays, and their marked difference from the essays of non-majors at least implies that there might be a real attitude difference caused by their experiences in college.

6. CONCLUSIONS

Our research suggests that building an introductory computer science course designed to be engaging, relevant to student interests, and focused on the practice of programming might not be a viable way to significantly increase interest in Computer Science as major. Perhaps a course emphasizing other goals might be more successful, or perhaps a single required course is just not enough cause a change in attitudes this significant. The difficulty of changing attitudes also suggests learning programming might not be the same as instilling computational habits of mind or giving them a useful understanding of the computational processes that affect their lives.

If teaching programming without changing student attitudes is a worthwhile goal, two non-major attitudes discussed here each have their own implications. It does not seem to be sufficient to present programming in an engaging way, if we hope to allay student concerns about relevance. Students seem capable of learning programming and then not using those skills further if they are not compatible with their overall computing goals. To promote programming (and Computer Science) without changing attitudes, we need to make real benefits of programming clear.

Does the fact the we are changing attitudes in computing majors tells us about changing the attitudes of non-majors? It may be that something as significant as a change in attitude can not happen in a single introductory course. Though the pre-college stories of computing majors was not significantly different than non-majors, there could be something that predisposed these students to computer science and made them more likely to think in this way. It may be that the fact that computing majors are committed to the major might make them more apt to change their attitudes. There is also the possibility that a curriculum focusing less on the specifics of programming (however relevant the context) might have a greater chance of changing student attitudes to something more in line with what majors seem to find most central about computer science. Whatever the source of this change, the fact that Computer Science is at least changing the attitudes of some of our majors is a hopeful sign for the potential of changing more student attitudes in the future.

7. ACKNOWLEDGEMENTS

This research is supported in part by a grant from the National Science Foundation BPC Program #0634629.

8. REFERENCES

[1] A. W. Astin. *What Matters in College?: Four Critical Years Revisited*. Jossey-Bass Inc., 1992.

[2] J. Bennedsen and M. E. Caspersen. An investigation of potential success factors for an introductory model-driven programming course. In *ICER '05: Proceedings of the 2005 international workshop on Computing education research*, pages 155–163, New York, NY, USA, 2005. ACM.

[3] M. Biggers, A. Brauer, and T. Yilmaz. Student perceptions of computer science: a retention study comparing graduating seniors with CS leavers. In *SIGCSE '08: Proceedings of the 39th SIGCSE technical symposium on Computer science education*, pages 402–406, New York, NY, USA, 2008. ACM.

[4] T. Camp. The incredible shrinking pipeline. *Commun. ACM*, 40(10):103–110, 1997.

[5] L. Carter. Why students with an apparent aptitude for computer science don't choose to major in computer science. In *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 27–31, New York, NY, USA, 2006. ACM.

[6] M. E. Caspersen, K. D. Larsen, and J. Bennedsen. Mental models and programming aptitude. *SIGCSE Bull.*, 39(3):206–210, 2007.

[7] M. Guzdial and A. Forte. Design process for a non-majors computing course. *SIGCSE Bull.*, 37(1):361–365, 2005.

[8] M. Guzdial and A. E. Tew. Imagineering inauthentic legitimate peripheral participation: an instructional design approach for motivating computing education. In *ICER '06: Proceedings of the 2006 international workshop on Computing education research*, pages 51–58, New York, NY, USA, 2006. ACM.

[9] F. N. Kerlinger. Social attitudes and their criterial referents. *Psychological Review*, 74(2):110–122, 1967.

[10] Q. H. Mahmoud. Revitalizing computing science education. *Computer*, 38(5):100, 98–99, 2005.

[11] J. Margolis and A. Fisher. *Unlocking the clubhouse : women in computing*. MIT Press, 2002.

[12] E. T. Pascarella and P. T. Terenzini. *How College Affects Students*. Jossey-Bass Inc., 1991.

[13] L. Rich, H. Perry, and M. Guzdial. A CS1 course designed to address interests of women. In *SIGCSE '04: Proceedings of the 35th SIGCSE technical symposium on Computer science education*, pages 190–194, New York, NY, USA, 2004. ACM.

[14] M. Rokeach. Attitude change and behavioral change. *The Public Opinion Quarterly*, 30(4):529–550, 1966.

[15] M. Rokeach. *Beliefs, attitudes, and values; a theory of organization and change*. Jossey-Bass, San Francisco, 1968.

[16] C. Schulte and M. Knobelsdorf. Attitudes towards computer science-computing experiences as a starting point and barrier to computer science. In *ICER '07: Proceedings of the third international workshop on Computing education research*, pages 27–38, New York, NY, USA, 2007. ACM.

[17] C. P. Snow. *Computers and the World of the Future*, chapter 1. MIT. Press, 1962.

[18] A. Strauss and J. Corbin. *Basics of qualitative research: grounded theory procedures and techniques*. Sage Publications, Newbury Park, Calif., 1990.

[19] A. E. Tew, C. Fowler, and M. Guzdial. Tracking an innovation in introductory cs education from a research university to a two-year college. In *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education*, pages 416–420, New York, NY, USA, 2005. ACM.

[20] J. M. Wing. Computational thinking. *Commun. ACM*, 49(3):33–35, 2006.