What Is Computing? Bridging the Gap Between Teenagers' Perceptions and Graduate Students' Experiences

Sarita Yardi, Amy Bruckman
College of Computing
Georgia Institute of Technology
85 5th Street, NW
Atlanta, GA 30332
{yardi,asb}@cc.gatech.edu

ABSTRACT

Studies show that teenagers perceive computing to be boring, antisocial, and irrelevant to their lives. We interviewed 13 teenagers from local Atlanta schools and observed over 40 teenagers in after-school technology programs to learn more about their perceptions of computing. We then interviewed 22 graduate students in the Human-Centered Computing and Human-Computer Interaction programs at Georgia Tech in order to learn about the factors that motivated them to pursue degrees in computing. We found that teenagers perceived computing to be boring, solitary, and lacking real-world context, yet graduate students described their research as exciting, social, and having a direct and meaningful impact on the world around them. Our results suggest that there is an opportunity to increase interest in computing among teenagers by bridging the gap between their perceptions of computing and the actual opportunities that are offered in computing disciplines. In this paper, we first describe our interview results. We then discuss our findings and propose a design-based curriculum to teach teenagers core computing principles. The goal of this curriculum is to prepare and motivate them for careers in today's expanding, Internet-based, global economy. We suggest that by portraying computing as an innovative, creative, and challenging field with authentic, real-world applications, we may be able to motivate teenagers to become more excited to pursue careers in computing.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer Science Education, Curriculum*; K.7.0 [**The Computing Profession**]: General

General Terms

Design

Keywords

broadening participation, teenagers, culture, HCI, computer science education, curriculum

1. INTRODUCTION

Over 87% of youth in America between the ages of 12 and 17 use the Internet and usage spikes from 60% in the sixth grade to 82% by seventh grade [12]. Despite their enthusiasm using these online tools and digital technologies, teenagers convey a significant lack of interest in computer science (CS) as an academic field or career path. CS enrollment rates spiked in the late 1990s when the "dot com bubble" offered computing jobs that were perceived as exciting, cool, and lucrative. However, there has since been a large drop in CS enrollment [25].

Our research is being conducted as part of Georgia Computes, an NSF Broadening Participation in Computing alliance to change the perception of computing by showing students that there are opportunities in computing that are exciting, innovative, and have real-world applications. We are exploring ways that educators can leverage students' existing areas of interest and expertise using computers and technology as a hook to bridge them into computing disciplines. In this paper, we describe the research we conducted with Atlanta teenagers and graduate students in the College of Computing at Georgia Tech. We first describe teenagers' existing perceptions of computing. We then draw from the stories and experiences of the graduate students to show the ways in which the field of computing is an exciting, challenging, socially-oriented discipline that can have broad impacts in the world around us. Finally, we describe our approach to engaging teenagers in computing through a design-based computing curriculum.

2. DECLINING INTEREST IN COMPUTING

The decline in participation in CS is attributed to students' perceptions of computing to be boring, asocial, and irrelevant to their lives [15]. These perceptions, as well as their beliefs that the opportunities for good careers in computing were lost in the dot com bust or are being migrated to developing countries, have led to a crisis in the field of computing. Students do not see computing as an environment for creativity [20] and believe that it focuses on tedious details instead of important topics in problem solving [14]. Furthermore, students from underrepresented groups often perceive themselves to be unwelcome in the field [14, 7], feel uncomfortable asking for help [27], and find the culture surrounding CS to be aggressive [3]. These perceptions have contributed to the drastic declines in participation in undergraduate and graduate CS departments. For example, the number of students who declared a major in CS decreased by almost 50% in a six year span (see Figure 1) [15, 26]. Similarly, overall enrollment rates in CS classes dropped to less than half of what they were at their peak in 2000 (see Figure 2) [26].

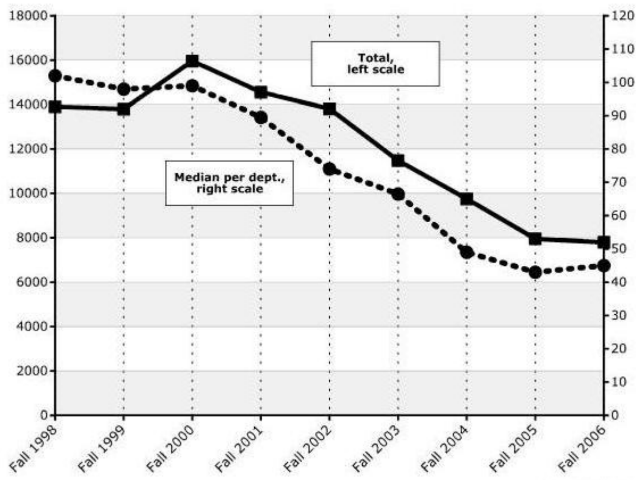

Figure 1: New CS Majors, Source: CRA Taulbee Survey

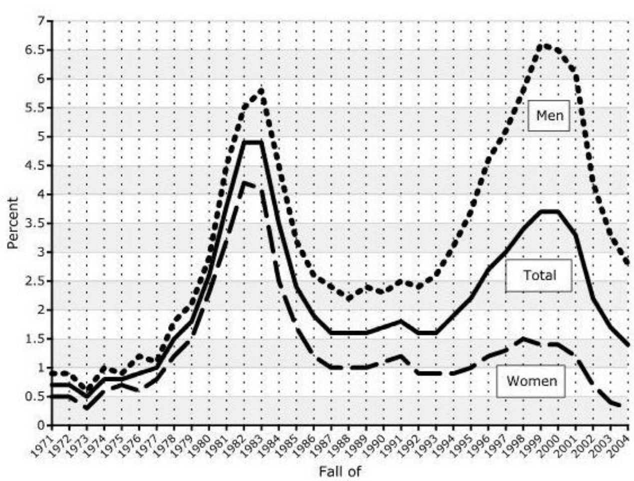

Figure 2: CS Listed as Probable Major, Source: HERI at UCLA

The problem of declining participation is not limited to higher education. Today's high school CS classes face multiple constraints that limit their effectiveness in teaching students required computing skills. First, schools struggle to gain state and federal support for CS curricula in schools. As a result, CS courses are usually offered as electives, and are therefore given much lower priority than the courses that are required for high school graduation [21]. A second problem is that few resources are committed to training teachers in CS. Most CS majors will choose a more lucrative career in industry over a teaching career, resulting in a shortage of knowledgeable CS teachers at the high school level. Furthermore, there are few or no CS Education degrees offered in higher education curricula that can teach the specific skills required for teaching CS. A third problem is that existing high school curricula are often out of date and frequently fail to teach basic computing skills, resulting in students having a minimal understanding of programming languages, weak problem solving skills, and difficulty in conducting analysis and design [24].

A recent ACM Task Force study states that: "The current study of computers in high school is characterized by either the use of computers as a tool for other disciplines. . . or programming. Neither of these approaches defines CS, although both comprise aspects of the discipline. The study of CS is composed of basic universal concepts that transcend the technology and that comprise an essential part of a high school education. It is these concepts that enable the student to understand and participate effectively in our modern world" [1]. As suggested in this report, there are core computing principles that teenagers can be taught, and these principles can be taught in a way that is engaging and motivating for them. We are looking to explore ways in which computing can be introduced to teenagers that will increase their levels of participation, be motivating for them, and can establish sustainable continuity in their computing education. Framed in this context, we are asking: How can we teach core computing skills to teenagers and how can we motivate them to pursue further computing related educational opportunities?

3. GEORGIA COMPUTES

The goal of Georgia Computes is to understand teenagers' perceptions of computing and how we might encourage them to be more interested in pursuing computing degrees. Although computing research in academia and computing practice in industry require different skills and interests, we chose to interview graduate students rather than undergraduates who might be planning to work in industry. In particular, the majority of graduate students have at some point made an explicit decision to pursue computing careers whereas undergrads might still be choosing between majors. Graduate students may therefore have an easier time describing their reasons for pursuing computing degrees and the decisions they made that led them there.

We interviewed graduate students in the Human-Computer Interaction[1] (HCI) M.S. program and the Human-Centered Computing[2] (HCC) Ph.D. program at Georgia Tech. Many of them came to these programs from non-traditional backgrounds and did not necessarily have early interests in math or science as traditional CS majors have had. We focused on students in these programs because both programs are relatively new models, with their own strengths and weaknesses, that can help to lend insights into a range of diverse computing experiences. A better understanding of the paths students in these programs took to end up in computing may help us to develop alternative methods for reaching out to younger students. Our future research plans including interviewing students in traditional CS disciplines to compare

[1]http://www.cc.gatech.edu/content/view/731/464/
[2]http://www.cc.gatech.edu/content/view/190/136/

experiences across traditional and non-traditional computing disciplines. It is important to emphasize that CS is not the same as HCI and that each field requires significantly different skills sets. Not all students who are interested in HCI will be successful, or interested, in becoming Computer Scientists.

The ACM's 2005 Computing Curricula Report describes the major disciplines in computing—such as computer science, information systems, and software engineering—and the types of skills that should be taught in K-12 to prepare students for careers in these disciplines [23]. Our goal is to encourage teenagers to consider pursuing these specific disciplines and to also suggest that there are core computing skills that all students should be taught as members of today's workforce. In the following sections, we describe existing perceptions and experiences in HCI and explore ways of teaching computing through an HCI model.

3.1 Interviews with Teenagers

We interviewed local Atlanta teenagers to learn more about their perceptions of CS and their everyday activities on the Internet and using other types of technology. We talked to 13 students overall, eight girls and five boys. Three were minorities and the remaining 10 were Caucasian[3]. Our study participants ranged in age from 11-20. Throughout this paper we refer to our interview participants as "teenagers". They ranged in demographics, with the majority being white, middle-class females.

We recruited participants using a variety of methods, including word-of-mouth, contacting high school administrative contacts, leveraging existing partnerships with after-school programs, and semi-random emailing. The variety in sampling methods helped us to recruit a diverse participant pool. We also conducted participant observations in informal technology clubs, observing 14 students in a high school digital media club as well as 28 girl scouts who were participants in robotics workshops. All of the students in the high school club were minorities while 15 of the 28 girl scout participants were minorities.

We conducted our interviews and focus groups with two particular questions in mind:

1. What are teenagers' current practices surrounding their use of computers, the Internet, and technology in their informal, everyday lives?

2. What are teenagers' perceptions of CS and computing related fields and do they see themselves pursuing degrees or careers in these disciplines?

Our purpose for asking these two questions was to help us explore and better understand our existing hypotheses about why students are not interested in CS. We also looked to understand what they did for fun in these environments to help us design an intervention that would meet them in the middle of these two spaces. The majority of the teenagers we talked to expressed a lack of interest in computing related careers, citing the field to be boring, difficult, and tedious. We present our interview results and analysis in Section 4.

3.2 Interviews with Graduate Students

We then interviewed 22 graduate students in Georgia Tech's College of Computing during the Spring of 2007. All students were currently enrolled in the M.S. HCI or Ph.D. HCC programs. The HCC Ph.D. program at Georgia Tech combines multiple disciplines, including HCI, learning sciences and technology, cognitive science, artificial intelligence, robotics, software engineering and information security. Its focus is on "developing theory and experimentation linking human concerns and computing in all areas of computing, ranging from the technical-use focus of programming languages and API designs and software engineering tools and methodologies to the impacts of computing technology on individuals, groups, organizations, and entire societies"[4]. The students we interviewed came from any of these disciplines as well as a range of others, and expressed a general sense of excitement, engagement, and committment to their courses and research within each of their specific fields of interest.

Table 1: Total Enrollment by Academic Year

	2005-2006	2006-2007
HCI M.S.	46	52
HCC Ph.D.	11	27

We contacted 54 students and received responses from 25 students. We were able to schedule interviews with 22 of these participants. We did not follow up with the remaining students because of high response rate (46%). It should be noted that our response rate can likely be attributed to our research group's presence within the College of Computing and that our interview topic was of general personal interest to many of the participants.

Of the 22 participants, 10 were female and 12 were male while nine were minorities and 13 were Caucasian. 15 were current HCC Ph.D. students while 7 were HCI M.S. students. Neither of these breakdowns was a statistically significant deviation from the overall HCC and HCI demographics (see Table 2).

Table 2: Enrollment Demographics

	Male	Female	Minority	Total
HCI M.S.	29 (56%)	23 (44%)	23 (44%)	52
HCC Ph.D.	11 (41%)	16 (59%)	12 (44%)	27
Interviews	12 (55%)	10 (45%)	9 (41%)	22

We conducted the interviews in a structured format with the purpose of understanding what factors motivated the students to pursue degrees in HCI or HCC and what early experiences they had that may have encouraged them to pursue these fields. In particular, we designed our interview protocol to investigate three specific questions:

1. What factors motivated students to pursue degrees and careers in HCI and HCC?

2. Did they have any anecdotes and stories from their childhood and teenage years that suggested an early interest in or predisposition towards these fields?

3. Did they have any advice for us in how to design an effective HCI curriculum for teenagers? In particular:

   (a) What would be the important skills that teenagers should learn?

[3]We use the term minority as defined by the US Census

[4]http://www.cc.gatech.edu/hcc/

(b) What would be an effective way to teach these skills?

(c) What project topics would be exciting or motivating for teenagers?

Our goal in asking the first question was to leverage graduate student experiences in order to learn more about how today's teenagers might be encouraged to pursue HCI and HCC related disciplines. Our goal in asking the second question was to gather anecdotes, stories, and narrative descriptions about students' early activities related to computing in their informal, out-of-school environments. The goal of the third question was to solicit advice, based on their own experiences as experts in HCI and HCC, to help us in designing our curriculum. We describe our interview results and analysis in Section 5.

3.3 Data Analysis

We used qualitative data software to code our interviews. Medley (2001) describes the importance of qualitative methods in computer science education, emphasizing the value that qualitative software can have for describing and analyzing interview transcripts [16]. We used Weft QDA[5], an open-source software tool for the analysis of textual data. We uploaded our interview transcripts into the software and coded our data by marking, or "highlighting", passages of text. We used the themes that emerged to generate a coding framework. We then conducted a second iteration of coding in which we coded the common key themes across each of the interviews. We grouped interview passages by theme and cross-tabulated multiple dimensions to compare teenager and graduate student experiences. In the following section, we describe the teenagers' perceptions of computing and the themes that emerged from our analysis.

4. TEENAGERS' PERCEPTIONS

Our interviews with the teenagers supported existing research that shows them to be creative, passionate, and enthusiastically engaged in their online environments [11]. Numerous large scale national studies have confirmed the increasing rate at which teenagers are online and their increasingly advanced and sophisticated practices in these environments (e.g. [11]). Teenagers are actively creating blogs and webpages, posting original artwork, photography, stories or videos online and most teenagers have accounts on MySpace, Facebook, or Xanga. They also keep in touch through a vast array of communication tools, such as instant messaging, email, message boards, or chat rooms. The average teen instant messages with over thirty of his or her peers for three hours a week [12] and most of the particants in our study had over 200 friends on Facebook. All of them had cell phones and reported text messaging multiple times a day. They also reported logging into a chat client, usually AIM, almost every evening after school. However, despite their enthusiasm using these online tools and digital technologies, the teenagers we interviewed conveyed a significant lack of interest in CS as an academic field or career path. They described the ways in which they perceived CS to be difficult, boring, and asocial (see Table 3). One participant stated that: "I don't think I want to be a programmer because it's too tedious and I don't think I could do that, sitting in front of a screen all day, just looking at the typed stuff, I don't do well." Another said "I'm more of a people person." A third stated that "It's cool. It's amazing how it all works. I could never do anything like that." Their image of CS as a field of programmers working in solitude is consistent with previous findings, in which people understood "programming" to refer to a discipline that was concerned with design, development, testing, debugging, documentation, and maintenance of software" [6]. When asked about a particular computing course, a junior high student declared that: "I didn't really find it very fun because I mean they basically made us do boring stuff like draw big, small circles and like big circles and they didn't exactly explain what was going to happen and stuff." When asked to define CS, one younger participant hesitated:

Table 3: Reasons for Lack of Interest in CS (n=13)

Boring	9
Only for the Smart Students	7
Antisocial	5
Lacking Creativity	4
Tedious	3

> I think it's a language... like different languages in computers and making a computer, like, change the sound or the visual stuff, like the different colors or something. Well, I know I'd have to do a lot about science 'cause you have to know stuff about like electricity and all these other things to make things like happen... I know to be a computer scientist you have to be a problem solver 'cause you have to change things in CS to make colors different. (Rich, 13)[6]

When asked what CS was, another participant stated that it involved "operating systems and web applications and like the skeleton of everything that goes in your computer. It's not necessarily what it looks like but like how it works." However, not all of the participants professed a lack of ability or interest in computing. We asked one high school female, who was highly active in online environments, if she thought her peers could become more interested in the field and why:

> I mean, most of the people I know have the capacity to know about computers as I do. I think a big part of it has to do with, um, like more of a social thing... I definitely see the people in the technology group as being anti-social when they were younger.... Now they're just as social as everybody else because in high school everybody's pretty much accepted..." (Jen, 16)

When asked about how to change the perceptions based on these stereotypes, another participant said:

> There are all sorts of big movements about [kids] in technology and scholarship and all that and that's really great that people are doing that but when I think about my friends who are the ones that have the capacity to learn really fast and could easily do this, they just choose not to, it

[5]http://www.pressure.to/qda

[6]Pseudonyms are used to protect participants' identity

> has to be on a smaller scale, a required computer course at school, an introduction to all the neat things on the web. (Josh, 16)

A number of our participants had taken some sort of programming classes, ranging from topics like digital design and computer animation to AP CS. They expressed a range of mixed experiences and perceptions of these courses.

> My dad promised me a car if I signed up for CS and I did CS 1 and never got a car but I fell in love with it so I did CS 2 and then I went into AP for a whole semester and then I decided that I didn't like AP so I did an Independent Study, which I'm doing now. (Nick, 15)

We also found that many of the female participants had strong perceptions of computing to be a male dominated field. One advanced student who had been the only female in a high school CS course stated that:

> When I went into [AP CS], [the other kids] were like 'You're in AP?' and I was like 'Yes, I am... ' [CS] is hard. I mean it's not anything easy. When I did AP, I was struggling within the first three weeks. [The other kids], they are fine, they can program, what would take me like two months to do, they can do in three weeks. (Leila, 16)

Many of the high school students demonstrated highly proficient and advanced understanding of online tools and environments, but were still very fearful of computing, programming, and the associated nature of those disciplines. When asked what they were good at on the computer, they stated: "You know, like use FaceBook, check your email", "use iTunes", and "like Microsoft Word and PowerPoint." However, when asked about CS specifically, one said: "I don't know of [anyone] that's like really good with it."

While our findings reinforced much of what we already knew about teenagers' perceptions of computing, they also offered us a much more detailed and in-depth understanding of their perceptions from their own eyes. We acquired a more complete picture of what environments they did enjoy, what they found fun, and what paths might be useful for us to lead them to become more interested in computing. After interviewing teenagers, our next step was to look to understand how their perceptions of computing compared to actual graduate student experiences in computing. In our graduate student interviews, we hoped to learn more about what motivated them to enter the field of computing and what their perceptions of computing were. In the following sections we first describe the HCI and HCC programs at Georgia Tech and then describe students' experiences in these programs.

5. GRADUATE STUDENTS' EXPERIENCES

5.1 Why Study Computing?

We found that many of our participants were not aware of the field of HCI and HCC until just before applying to the program. Eight had entered graduate school directly after getting their undergraduate degrees while the remaining 14 had returned to graduate school after working in a variety of industry jobs. Regardless of previous experiences, all participants described experiencing excitement and surprise when they learned that there were programs in HCI and HCC. Their excitement was usually rooted in the nature of these programs as interdisciplinary and offering the opportunity to combine their multiple, diverse interests into a single field of study. Our coding of the data revealed a number of key themes: interdisciplinarity, creativity, parental influence, games, and fun (see Table 4).

Table 4: Reasons for Studying HCI or HCC (n=22)

Real-World Relevance	18
Interdisciplinary Field of Study	12
Opportunity for Creativity	5
Early Role Models	5
Interest in Games (Video, Console, Computer)	3
Projects are Fun	2

5.2 From "How Does This Work?" to "Why Is This Useful?"

Many participants expressed a peripheral interest in programming and CS, but chose HCI and HCC because they were looking for a more interdisciplinary, multi-faceted experience. For example:

> I guess a lot of times in CS I realized I was different from other people in the way I was thinking, a lot of people think things are cool just for existing. For me, the issue was who would use it? In a lot of CS classes, something new would be presented and most people's question was 'how does it work?' Mine was 'what is it useful for?'

Similarly, another was excited about the opportunity to combine his interest in visual arts and computing:

> I was a total nerd growing up... I was always into the visual aesthetics of things. I found myself doing more and more development, more programming. HCI was kind of a cool way to help me balance my design skills and programming, it was a cool merging of art and technology. I started to see the art behind computing and I definitely saw art behind visual functionality.

In their interviews, many participants first told us about their undergraduate experiences then highlighted the ways in which the undergraduate degree fulfilled some of their academic interests, but were too focused on particular areas of exertise.

> It was kind of cool to combine some technology and some other fields like psychology, culture, and communication and to understand it better—how people can use technology. I took some computation during undergrad where you have to try to make the car move and you need to program some special software. During the summer we got some training and we finally got our car to move and could control it. It made me feel like I can do that... it was cool.

> As an undergrad, I was wondering what are the research groups on campus. I saw HCI and thought holy [****], they are doing the stuff we were talking about [as teenagers] but just not understanding what we were talking about [at the time]!

In the following section, we describe the graduate students' childhood stories, many of which were precursors to their entry into the field of computing.

5.3 Gears, Clocks, and Games: Capturing Early Experiences

In *Gears of My Childhood*, Seymour Papert describes his childhood love for gears starting at the age of two and how this early fascination later helped him to model more abstract ideas about cars and the learning process in general [18]. Although many people may not have such profound memories of early childhood experiences, most can remember stories about childhood activities or interests that sparked later interests through their teenage and childhood years. In our graduate student interviews, we encouraged participants to reflect back on their early experiences and to try to recall anecdotes that may have led them to become excited about computing related fields in later years. Although a limitation in this method is that participants are asked to describe experiences that will be inevitably constructed and reconstructed within the framework of their existing experiences, they were able to describe early memories that correlated to their existing interests.

The majority of the participants mentioned having some exposure to computers as children or teenagers and had specific memories about the ways in which the computer began to play a role in their lives (see Table 5). They described how their hobbies had an early influence on their interest in technology:

> I always enjoyed playing with Legos. Legos were a big part of my childhood. Whenever I accomplished something significant my parents would get me a Lego set and I would build something new. Eventually I amassed a huge collection of Legos... That got me interested in the idea of building 3d objects that I wouldn't be able to make in real life. From there I started messing around with 3d animation on the computer and building things.

Table 5: Childhood Stories (n=22)

Interest in Games (Video, Console, Computer)	8
Early Role Models (Parents)	8
Opportunity for Creativity	7
Interdisciplinary Field of Study	4
Use of Multimedia	3
Projects are Fun	3

While hobbies, games, and video games in particular were popular, there were other applications like PowerPoint and Photoshop as well as activities related to art or school that influenced their engagement with computers. However, not all participants expressed an inclination for working with their hands, hacking hardware or software, or exploring unfamiliar environments. Others described taking apart objects around the house, such as clocks or the coils on a kitchen stovetops, because they were curious to learn about how they worked. Some told us that they had no early interests in technology of any sort and instead were motivated by intellectual challenges of computing or the social and communication oriented possibilities that computing fields offered. We gathered a wide range of stories among the 22 participants, in which some recalled ongoing hobbies that they felt undoubtedly influenced their love for working with their hands and creating technologies. Other participants mentioned particular events in which they recalled having a sense of curiosity about how something worked, or taking something apart to explore its inner functionality.

6. DISCUSSION: BRIDGING PERCEPTIONS OF COMPUTING

How can we attract teenagers to the field of computing? What are the key factors that will engage them by breaking down their existing perceptions of computing? Our interviews affirmed many existing studies about what types of computing environments might help to broaden the range of students in the field [14, 4]. In our discussion, we look to address each of the following questions in more detail based on our research:

- Why are teenagers so excited about using computers as socially oriented communication tools yet so averse to using computers for CS?
- What contexts will motivate and excite teenagers to further pursue computing careers?
- What are the prerequisite skills that will effectively prepare teenagers for careers in computing?
- What actionable steps can the CS education communities take towards realizing this agenda?
- What are the fundamental HCI principles that we can teach them and how can we ground these principles in projects that have socially relevant applications?

We are looking to explore the possibility of leveraging teenagers' existing interests to engage them in computing activities. We compared the most common themes across the teenagers' and graduate students' responses, looking for specific overlaps in topics. We found that many common themes emerged from their stories. The most frequent themes were related to real-world relevance, games, role models, and creativity. In the following section we discuss each of the themes in more detail.

6.1 Theme #1: Real-World Relevance

The importance of being able to apply HCI and HCC research to real-world problems was emphasized by most of the graduate students. They spoke of the social, communicative, and creative processes involved in their computing research areas, describing how they felt there was a purpose or need for what they were working on. One HCI student explained that she "got involved with [a university] music program and while I was there, I started seeing that there weren't a lot of education applications for the music world. I saw the HCI program, and found a way to make an impact in the classroom." Other students affirmed the importance of "gaining practical knowledge" and "being able to use" what they were learning in "real life". Another graduate student described his early interest in design that stemmed from his interest in cars:

> I followed automobile designs, as a kid I subscribed to auto magazines and looked at the cool shapes. Around the same time, Apple was getting notoriety for design. Apple was a big influence... I took an art class my senior year, but it didn't dawn on me I could do it for a living until year two.

In contrast, the teenagers we talked to often perceived little practical meaning or purpose behind the stereotypical CS profession. Many were unaware that many of their existing practices in their online activities could in fact be equipping them with important skills that would help them in the new wave of exciting careers in computing. Others expressed a desire to learn practical skills that could help them in their day to day practices. For example, one participant expressed an interest in wanting to understand how computers work: '

> My family is really stupid when they use computers. They break a lot. I wish I knew how to fix them so we wouldn't have to spend as much money for people to fix them. I wish I knew more about actual computers so I could fix them.

By expanding students' perceptions of the possible career options that their CS degree will offer, they may be more likely to perceive the major to be a viable path towards their own career goals.

6.2 Theme #2: Games

Many students who were in the College of Computing at Georgia Tech had not taken CS in high school. Instead, they expressed an interest in CS because they were interested in video games, cell phones, digital audio players, or the role of technology in other disciplines, such as biology [8]. In our interviews, for example, one HCC student who was interested in video game design said:

> I've always been into video games, which were what got me into the computer. When I was in 4th grade, I was like the local software dealer. I'd have a list of games and would trade one copy for another copy with the other kids.

He is now working towards his Ph.D. in game design studies. Another HCI student is designing video games for his M.S. thesis and described his early interest in games: "When I was younger, I wanted to make games. Every kid likes playing video games. We didn't have many video games and I wished I could make my own games. Then I got a computer and I made simple games" and similarly, "I enjoyed spending time with computers and particularly enjoyed playing games on the computer. It was the sort of thing I could do all day" and a third said:

> I think, also, like just games was a huge bridge for me, I started out playing atari games and nintendo games and saw that people had commodore games and nintendo 64. You had to know how to run your floppy drives to play those games, once I did that and once I had a computer, I learned about word processing and how to use a printer.

Both male and female graduate students were interested in games and other multimedia environments. These interests also led them towards a diverse set of research interests, ranging from virtual reality to artificial intelligence to information visualization.

6.3 Theme #3: Early Role Models

Most early role models were parents and friends. Many graduate students emphasized that their parents had either provided an opportunity to play with some sort of technology or that parents had been the focus of various childhood activities. One female said that she

> liked playing on the computer. I would just do PowerPoint presentations. I would make family PowerPoints, like of my mom, and would use a cartoon picture of her. I would just keep doing that and messing around. I liked doing that and wanted to learn more of it. I decided this was what I wanted to do.

Another parental influence was more direct, in which an HCC student's mother encouraged him to read about computers.

> [As a kid], my mom gave me books on computers, a 'kids intro to computing' text. It told you a lot about how computers worked. It was a well done book and sparked my interest in computers. As a kid, it was new to me, I didn't have access to a computer. I had video games, which were totally sweet. In high school, I started reading programming books. I guess I had a lot of free time.

A third student described his early memories of the Internet and at what point his parents finally gave in and set up a modem for him:

> I remember arguing with my dad - I was kind of a rotten kid - because we didn't have a modem and didn't have service. I knew about the Internet and my friends had the Internet... I should be able to just plug it in and type in something.

Another said: "I always had an inclination towards computers, probably because of my dad." Many graduate students who had parents in computing related professions were especially likely to mention very young experiences with computers and to have had a greater sense of familiarity and comfort in these environments.

6.4 Theme #4: Creativity

Many graduate students chose their research because it allowed them to think creatively and explore innovative ideas. The teenagers reinforced these notions through their lack of interest in CS because they did not believe it to offer any opportunities for creativity. For example, one teenage girl told us she was interested in search and retrieval websites as well as technology law and policy. She later said: "I love [Google's] philosophy... they know everything about me. It kind of scares me how much they know about me, they save everything. They've been really good about their policies so far when asked to give up their information in not doing that, but still, the company has to change hands at some point. . . . This same participant stated that "I decided in like 7th grade I wanted to work for Google" but later declared that she did not want to go into programming, not because it was too challenging, but because "I don't think I want to be a programmer because it's too tedious and I don't think I could do that, sitting in front of a screen all day, just looking at the typed stuff, I don't do well."

In contrast, a graduate student said that: "It wasn't only about the ideas, but the process of creating something. I wanted to be a carpenter. CS was only thing with science that involved creating something." Similarly, another said "I've always enjoyed making things. . . I always enjoyed making things and not just Lego making things. I used to experiment with Play-Doh. . . I would experiment with what would make it last a long time." One student summarized her experinces especially well: "HCI lets you do a whole lot of things, I guess I was always interested in a lot of areas. There is a wide variety of opportunities. It's always fun. It's always on the cutting edge. There are always interesting things going on."

6.5 Synthesizing Themes

We believe that we can leverage these existing interests to engage teenagers in contextualized computing projects where they can apply solutions to real-world problems and can see the authenticity and importance in their work. Researchers in higher education have been increasingly recognizing the importance of personally meaningful projects and have seen successful results in initiatives that are designed around this goal in higher education. For example, in a social networking based "gateway course" at Cornell University, school administrators expected 15 students to enroll, but 220 signed up for the class. "Instead of doing all the fundamentals first and then getting to the exciting parts just flip it over so the students see what the exciting parts are and then they are motivated to learn the fundamentals" [22]. A similar class at Duke University taught programming principles through social networking, looking to engage students by building on the popularity of existing social networking sites [2]. Although the same fundamental computing principles are being taught to students in these courses, the context in which they are being taught is more relevant and applicable in their lives. "We'd seen very narrow CS departments graduating students focused only on specific kinds of technologies, and frankly, it is the least exciting part of the field. . . . We tried to focus on what exciting things people do with technology and see if it changed the fields. People start to pour into programs when they think technology is being used for something relevant and fun to them" [19].

Based on our analysis, we suggest that the following three factors should be included in a computing curriculum in order to engage students:

- The curriculum should be personally relevant to the student.
- Students should be able to recognize a specific purpose for working on an assignment.
- Students should perceive a real-world implication or relevant social application for an assignment.

Why are these findings important for computing education researchers? Our research suggests an opportunity for CS educators to leverage students' existing areas of expertise in computing and technology environments as a hook to bridge them into CS and programming. In the following section, we propose a particular curriculum that incorporates these principles and discuss ways of implementing it in the classroom.

7. IMPLICATIONS FOR CS EDUCATION

We are creating an "Introduction to HCI for Teenagers" curriculum in which teenagers will be taught basic HCI principles in the context of a design-based project. We will be conducting a pilot elective course during the summer of 2007 with Atlanta public school students. The course will be offered as a voluntary elective with a two-month long summer school program that they are participating in. The curriculum will be loosely modeled on college level HCI and User Interface Design type courses[7]. We designed our syllabus to incorporate key themes that were recommended by current HCI and HCC graduate students as motivating factors for encouraging teenagers into the field (see Table 6).

Table 6: Advice for Teaching HCI (n=22)

Projects Should Be Fun	8
Pull From Multiple Disciplines	7
Use Multimedia	6
Integrate Games	4
Provide Opportunities for Creativity	3

While a typical college level introductory HCI course covers about 10-15 core lessons, including topics such as affordances, mappings, wire framing, and usability testing, our curriculum is designed to emphasize two main high level skills that are the foundation of most of these core lessons: principles of user interface design and user studies methods. We are focusing on these two items with a higher level of granularity, looking to cover important topics in depth rather than a broad range of topics (see Table 7).

Our goal is to emphasize the core concepts and reinforce them over multiple lessons in order to make sure teenagers grasp the fundamental principles as well as to ensure that they leave the course feeling confident that they have acquired key skills that the course was designed to teach them. Students with more experience using computers may also come away with the skills to apply the principles and methodologies related to HCI in both computing and other disciplines. More advanced students can also learn to develop algorithmic thinking skills and problem-solving skills and to use the skills in other applications. Particular principles and methodologies, such as interface design principles, designing for the user, prototyping, and working within teams, can be broadly applied across disciplines and in almost any future career path that the students might take. This broad application fits well into the the ACM Computing Board's emphasis on the importance of CS "in the modern world", which states that the goal is to "help students to use computers effectively in their lives, thus providing a foundation for successfully integrating their own interests and careers with the resources of a technological society." Our partici-

[7]e.g. http://courses.ischool.berkeley.edu/i213/s07/, http://www-static.cc.gatech.edu/ john.stasko/6750/

Table 7: Sample Skills and Activities

User Studies	Interview other teenagers about use of related interface designs.
Prototyping	Create paper and pencil prototypes. Refine and iterate. Create prototypes using web-based software tools.
Modularity	Create MySpace code modules that other students in the class can reuse.
Data Structures	Search through linked list of friends and display profile pictures.
Design Tools	Drag and drop web-based tools to lay out interface design.
Collaboration	Use group design tools for real-time, synchronous interface design.

pants expressed an overwhelming inclination towards acquiring more multimedia related skills on the computer. When asked, "If you could learn something new on the computer, what would it be?" they responded: "probably like GarageBand. I think it's cool how you can make music" and "editing videos and stuff" and "iTunes and Photobooth are really cool" and "I would want to learn how to fix the Internet." We will therefore encourage them to pick their own project, such as redesigning a Google interface for teenagers, writing a Facebook application, or prototyping a new multiplayer game. We are not emphasizing a specific technical skill but are instead looking to show teenagers that the field of computing is relevant to their lives. Participants will be encouraged to choose projects that are important and meaningful to them.

Students who have backgrounds in programming will be encouraged to choose projects that incorporate their skills into their design process, such as using a Facebook API to design a novel social networking application. Students who have little or no technical background and may feel intimidated in this environment will be encouraged to explore projects that allow them to explore a social, human, or service oriented approach to technology, such as designing a health related application for a grandparent with a particular illness. The syllabus is designed to plant a seed of interest in the role of computing in their lives by meeting them at their existing levels of comfort and familiarity using computers and technology.

They will be able to upload their projects to an online community that consists of their peers, graduate HCI and HCC students at Georgia Tech, and a general public audience. This online environment is designed to foster a sense of importance and relevance in their activities by linking them to a community of researchers and professionals who are implementing related real-world applications. This environment will portray graduate student experiencesm which can offer an opportunity for them to serve as role models for younger students. Instead of telling teenagers about the fun opportunities in computing, we can show them the projects that graduate students are working on, giving them a real-world and authentic reason to become more interested in the field.

7.1 Why Design?

We do not suggest that the process of learning design skills is the same as that of learning programming skills. However, we do suggest that the overlap in required skills and possible career opportunities offers design as a method for sparking an early interest in computing. A "design first" sequence helps to target and hook those students who might otherwise be averse to CS and discard the discipline before exploring its many facets. The theory behind this motivation is not new. Related research has suggested that a "games first" approach in a CS1 course resulted in higher retention and had a positive influence on both women and men, without any loss in the technical skills [13].

Our reasons for choosing a design-based curriculum are not only because it offers an engaging and fun way to introduce computing into student's lives, but that it is also highlighted as one of the key skills that should be taught to students as part of their computing curriculum [17, 5] and is a fundamental process within the field of HCI (e.g. [23]). Some software industry statistics have suggested that about 70% of a final design solution is determined during the early design stages and that about 50% of the overall development process is driven by the interface design [9]. However, many existing introductory CS classes do not teach within these contexts, instead focusing on repetitive, unoriginal problem sets that have few or no external real-world applications (e.g. [10]).

Despite the many advantages of using design-based processes in software creation and group projects, studies have shown that introductory programming courses make little progress in teaching analysis and design effectively, suggesting that the ways in which designers design, outside of programming environments, require higher order thinking processes than are needed for coding [24]. The challenges in incorporating these design processes into teaching programming are that students are generally taught to organize their code into modules, using classes and typical object oriented structures, without placing substantial thought in why a particular module should be designed as such and how it fits into the overall system [24]. By incorporating design into the computing curriculum, students can be exposed to a broader perspective of CS, develop skills that will guide them towards careers in computing fields such as software engineering, artificial intelligence, and HCI, and can be given more realistic, interesting, and personally relevant programming design activities [24].

7.2 We're Hiring! Emerging Careers in Computing

Our goals are to encourage teenagers to pursue the numerous careers that are emerging in today's expanding, Internet-based, global economy. Georgia Tech's HCC Ph.D. and HCI M.S. are examples of programs that are trying to address the demand for highly skilled, specialized students. Employers are looking for students who are skilled in not just technical domains, but in the social, user-centered, and creative aspects of computing.

We suggest that exposing students to these skills at an early age, during their junior high and high school years, may help them to pursue such fields in high school and college. The key skills that students should come way with include the ability to think creatively and innovatively, communicate, collaborate, conduct research, and to think critically, solve problems, and make decisions. We want students to be well-prepared, and more importantly, excited and enthusiastic about pursuing the careers in computing.

Opportunities at companies like Yahoo!, Google, and Microsoft reveal that there is a demand for these skills. For example, "Google's mission is intentionally broad and we actively explore interesting ideas. . . We are looking for bright, creative, and talented individuals who enjoy doing the research, engineering, product development and product management to transform ideas like these into products that are not just beneficial, but essential to millions of people every day[8]". Along the same lines, recent posted job opportunities at these companies include titles such as: Web Specialist, Software Engineer, User Experience Design, Lead Animator, Game Designer, and Product Designer. What do these jobs have in common? They require a set of skills that extends beyond the traditional requirements in CS.

The teenagers we interviewed expressed interest in working at companies like Facebook, Google, and YouTube, perceiving the cultures to be somehow different than those of traditional software companies. When asked about jobs relating to computing that they might find most exciting, they often responded with jobs that related to their own current interests and activities: "I don't really want to do much on computer animation but I thought it'd be cool to make a video game cause that'd be really cool to design how it is going to work and like all the different strategies and then, like say what you want, how you think they should change the person and all that stuff." We hope to teach students these skills to prepare them for what we suggest will be a growing field, with opportunities for creativity, innovation, intellectual challenge, foresight, and flexibility. We want to set the stage to prepare them to walk into any job interview, with Yahoo!, Google, or Microsoft, and have the technical, social, and communicative skills required for these jobs.

8. FUTURE WORK

Our research has shown how teenagers' perceptions of traditional CS careers present a significant barrier to increasing participation in these fields. The field of computing as it exists today is in need of a makeover. As noted before, we did not interview students in traditional CS disciplines, such as theory, networking, or databases. A important area of future work is to compare our study results to student experiences in these disciplines, which might offer additional insights into alternative approaches for capturing young students' interest in a variety of computing topics. Computing is more broad than CS, programming, or software engineering and educators may need to revamp existing perceptions of computing to reflect this broader definition.

8.1 The New Face of Computing

What does the new face of computing look like? Is a User Interface Designer at Google part of the new face of computing? Is a Video Game Designer at Microsoft a computer scientist? The emerging globalized economy suggests a need for revisiting the meaning of these fields. What characteristics would a new face of computing contain? Two key emerging skills and opportunities within computing are innovation and creativity. The ACM Education Board has emphasized the importance of advancing innovation in CS curricula in order to improve its image and appeal, identifying innovation as one of four core practices of computing [5]. As part of their Great Principles of Computing project, they are looking to show that innovation can be learned and a "spirit of innovation can permeate" throughout CS courses and curricula:

> Starting early would help win new hearts and minds to computing. We believe the prospect of participating in and causing innovations is highly appealing to students who want to make a difference in the world. How can this be done? The first challenge is to embed the foundational practices of innovation into the curriculum, so that students learn innovation by doing, without necessarily being aware they are engaged with systematic processes [6].

By embedding innovation and design activities into computing curricula, educators can engage students through more creative, meaningful, and exciting projects. Students themselves indicate an interest in these types of computing classes. The Fall 2007 incoming HCI M.S program at Georgia Tech will be its largest ever, with 39 new students. Junior high and high school students may also be similarly interested in HCI related programs if they are give the opportunity to participate in them at an earlier age. Our research suggests that implementing this new curriculum may help to reverse the existing crisis in computing by attracting students at an early age and retaining them through relevant and meaningful computing projects.

9. ACKNOWLEDGMENTS

This research is supported by a grant from the National Science Foundation BPC program #0634629. We thank the teenagers and graduate students who willingly volunteered to participate in our study.

10. REFERENCES

[1] Acm model high school computer science curriculum. Technical report, Task Force of the Pre-College Committee of the Education Board of the ACM, 1997. Available at http://www.acm.org/education/hscur. Accessed April 2000.

[2] C. Alt, O. Astrachan, J. Forbes, R. Lucic, and S. Rodger. Social networks generate interest in computer science. *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 438–442, 2006.

[3] L. J. Barker, K. Garvin-Doxas, and M. Jackson. Defensive climate in the computer science classroom. *SIGCSE '02: Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, pages 43–47, 2002.

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

[5] P. J. Denning. Great principles of computing. *Commun. ACM*, 46(11):15–20, 2003.

[6] P. J. Denning and A. McGettrick. Recentering computer science. *Commun. ACM*, 48(11):15–19, 2005.

[7] A. Fisher and J. Margolis. Unlocking the clubhouse: The carnegie mellon experience. *SIGCSE Bulletin*, 34(2):79–83, 2002.

[8]Excerpt from Google recruiting email to researcher in March, 2007

[8] M. Furst, C. Isbell, and M. Guzdial. Threads: how to restructure a computer science curriculum for a flat world. *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education*, pages 420–424, 2007.

[9] J. Hailpern, E. Hinterbichler, C. Leppert, D. Cook, and B. P. Bailey. Team storm: Demonstrating an interaction model for working with multiple ideas during creative group work. In *C&C '07: Proceedings of the 7th conference on Creativity & cognition*. ACM Press, 2007.

[10] L. Layman, L. Williams, and K. Slaten. Note to self: make assignments meaningful. *SIGCSE Bull.*, 39(1):459–463, 2007.

[11] A. Lenhart, M. Madden, and P. Hitlin. Teen content creators and consumers. Technical report, Washington D.C.: Pew Internet & American Life, July 2005.

[12] A. Lenhart, M. Madden, and P. Hitlin. Teens and technology: Youth are leading the transition to a fully wired and mobile nation. Technical report, Washington D.C.: Pew Internet & American Life, November 2005.

[13] S. Leutenegger and J. Edgington. A games first approach to teaching introductory programming. In *SIGCSE '07: Proceedings of the 38th SIGCSE technical symposium on Computer science education*, pages 115–118, New York, NY, USA, 2007. ACM Press.

[14] J. Margolis and A. Fisher. *Unlocking the Clubhouse: Women in Computing*. MIT Press, Cambridge, MA, 2002.

[15] A. McGettrick. The current crisis in computing: what are the real issues? *SIGCSE Bull.*, 39(1), 2007.

[16] M. D. Medley. Using qualitative research software for cs education research. In *ITiCSE '01: Proceedings of the 6th annual conference on Innovation and technology in computer science education*, pages 141–144, New York, NY, USA, 2001. ACM Press.

[17] NSF. Science of design program solicitation. Available at http://www.nsf.gov/.

[18] S. Papert. *Mindstorms: Children, computers, and powerful ideas*. Basic Books, January 1981.

[19] D. Perelman. How to stop the dilbertization of it. *eWeek.com*, 2007.

[20] S. L. Pfleeger, P. Teller, S. E. Castaneda, M. Wilson, and R. Lindley. Increasing the enrollment of women in computer science. *SIGCSE '01: Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, pages 386–387, 2001.

[21] R. Raskin. Do we need a national computing curriculum? March 2007. Available at http://tech.yahoo.com/blogs/raskin.

[22] T. Sanders. Computer majors down to bits, cornell sees decline after dot-com bust. *The Ithaca Journal*, 2007. Available at http://www.theithacajournal.com.

[23] R. Shackelford, A. McGettrick, R. Sloan, H. Topi, G. Davies, R. Kamali, J. Cross, J. Impagliazzo, R. LeBlanc, and B. Lunt. Computing curricula 2005: The overview report. In *SIGCSE '06: Proceedings of the 37th SIGCSE technical symposium on Computer science education*, pages 456–457, New York, NY, USA, 2006. ACM Press.

[24] J. Sims-Knight and R. Upchurch. Teaching object-oriented design without programming: A progress report. *Computer Science Education*, 4:135–136, 1993.

[25] J. Vegso. Interest in cs as a major drops among incoming freshmen. *Computing Research News*, 17(3), 2005.

[26] J. Vegso. Continued drop in cs bachelor's degree production and enrollments as the number of new majors stabilizes. *Computing Research News*, 19(2), 2007.

[27] B. C. Wilson and S. Shrock. Contributing to success in an introductory computer science course: a study of twelve factors. *SIGCSE '01: Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education*, pages 184–188, 2001.