Perceptions of Computer Science at a South African university

Vashti C. Galpin *, Ian D. Sanders

School of Computer Science, University of the Witwatersrand, Private Bag 3, Wits 2050, Johannesburg, South Africa

Abstract

First year students at the University of the Witwatersrand in Johannesburg, South Africa were surveyed about their perceptions of Computer Science before and towards the end of their first year courses. The aim of this research was to investigate how the students' attitudes changed during these courses and to assess the impact of the innovative breadth-first curriculum that has been developed in the School of Computer Science which emphasises the fundamentals of the discipline and the mathematical nature of Computer Science. The results show that most perceptions did not change much or that there were changes in both directions. More students, and particularly female students, were positive about their own understanding of the nature of Computer Science after the course than before. However, when asking specifically about jobs and course content, there was not a substantially deeper understanding at the end of the course of what content they would expect to encounter in a Computer Science course or working as a Computer Scientist. Fewer students, particularly male students, thought that Computer Science and mathematics were closely related after the course than before and this was an unexpected result, which may be the result of discrete mathematics topics being taught in courses separate from those in which continuous mathematics topics are taught. Students became less positive about working with computers after the course, a result which supports prior research, but is an issue for concern as computers will play some role in their future careers. The students found the courses challenging and different from their expectations with few students finding the courses unenjoyable.

Key words: post-secondary education, gender studies, country-specific developments

* Corresponding author

Email addresses: vashti@cs.wits.ac.za (Vashti C. Galpin), ian@cs.wits.ac.za (Ian D. Sanders).

URLs: http://www.cs.wits.ac.za/~vashti (Vashti C. Galpin), http://www.cs.wits.ac.za/~ian (Ian D. Sanders).

Preprint submitted to Elsevier Science *14 February 2006*

1 Introduction

This paper describes research into changes in first year Computer Science students' perceptions of and attitudes to Computer Science at a South African university, the University of the Witwatersrand (Wits), located in Johannesburg. Our experience and prior research (Herbert, 2000; Sanders and Galpin, 1994) have shown that students often only have a narrow understanding of the nature of Computer Science, even when they have chosen to study it, and that they often make a strong link between Computer Science and programming, to the exclusion of other aspects of the discipline. This is most likely due to lack of career counselling, and an introduction to computing at school level that focusses on applications, such as word processing, and programming. Another possible explanation for lack of understanding is the diversity of the discipline, since it contains aspects of science, engineering and mathematics (Denning, Comer, Gries, Mulder, Tucker, Turner et al., 1989) and because it is not well-defined (Nielsen, von Hellens, Greenhill, & Pringle, 1998).

Other research has also shown incorrect perceptions such as that computing careers only involve programming, there are limited career opportunities, there is little interaction with people, and that computing careers are linked to administration or secretarial work (Greening, 1998; Durndell and Thomson, 1997; Craig, 1997; Teague and Clarke, 1991; Clarke and Teague, 1996; Ryan, 1994; Symonds, 2000).

The research was also motivated by our innovative curriculum (Sanders and Mueller, 2000) which was developed to cope with the fact that some of our students have good school-leaving Mathematics marks, but no or little exposure to computers due to the legacies of apartheid. The curriculum positions Computer Science within the Mathematical Sciences and has a focus on fundamentals of the discipline, as opposed to specific programming languages or computing technology. The aim of the curriculum is to broaden students' understanding of the discipline and to use the mathematical aspects of the discipline as an entry point to the discipline.

The research investigated students' attitudes at the beginning and towards the end of our first year courses. The methodology used in this research involved comparing responses from questionnaires completed at the start and near the end of the academic year. For educational research, it is often difficult or impossible to sufficiently isolate various variables sufficiently to show causality. McMillan and Schumacher (2001) suggest that when extraneous variation cannot be controlled, then this needs to be taken into account in the interpretation of the results. In our case, the students are exposed to many influences during the year that may affect their perceptions, including our course. Since it is not possible for pragmatic and ethical reasons to present two different courses to them, or to remove factors other than the course, we have to consider the students within their environment, and be cautious in our interpretations.

The research shows that our students became more confident about their understanding of the nature of Computer Science, although when questioned about what content could be expected in Computer Science, did not show much change. The students changed their opinion about the relatedness of Computer Science and Mathematics, becoming less convinced of a link between the two. They also became more negative about working with computers, a concern since their future careers will most likely involve computers to a substantial extent. These results, as well as the fact that there was little change for many attitudes, raise questions about what influences students the most – what is said in lectures or what they spend time on in the laboratories, or external factors not related to the course. Additionally, the location of material may affect their perceptions – if discrete mathematics topics are only covered in Computer Science, then the link between this course, and mathematics courses based on continuous mathematics may be opaque to students.

In the rest of the document, the phrase "understanding of Computer Science" will be used to mean "understanding of the nature of Computer Science" as opposed to meaning success in learning the material in our courses. Similarly, "understanding the content of Computer Science" will be used to mean being able to identify different aspects of Computer Science and it will not mean having an understanding of the course material.

The structure of the document is as follows: the next section considers related research into attitudes, as well as giving details of the structure of the South African education system and our curriculum. Then we present the research methodology, the results and finally provide an interpretation and discussion of the results. The questionnaire used in this research can be found in a technical report (Galpin and Sanders, 2005) which presents a preliminary analysis of the data gathered by this questionnaire including some analysis not directly relevant to this paper.

2 Related research and background

Over the last three decades, there has been much research into attitudes to computers, computing and Computer Science. In this section, research investigating changes in attitudes is considered, together with research into understanding of Computer Science and computing careers, as well as gender differences in attitudes. Since the focus of this research is first year at university, different approaches to introductory Computer Science courses are described. Next, some information specific to our country and institution is given, namely, the structure of the education system in South Africa, and the curriculum taught at the University of the Witwatersrand.

2.1 Changes in attitudes

Previous research has considered the effect of courses on student attitudes in different countries. In South Africa, Finnie (1987) investigated attitudes using Lee's attitude assessment instrument (Lee, 1970), before and after a university first course in business computing which included the use of a financial modelling package and COBOL programming. After completion of the eight-month course, students became more negative about the value of computers and their impact, students were less in awe of computers and had a greater technical appreciation. Finnie notes the difficulty of implementing a control group who would not be exposed to the aspects of the course. A second South African study measured attitudes before and after a Computer Literacy course at another university (Burger and Blignaut, 2004). After the course, students were more anxious, were more negative, liked computers less and had less confidence in their abilities. The authors hypothesised that these effects may be reduced as students gained more experience.

Staehr, Martin, & Byrne (2001) compared attitudes before and after an introductory computing course at an Australian university. They found the students' liking of computers decreased by the end of the course. They speculate that this could be "due to over exposure to computers or the competition for scarce resources" (Staehr et al., 2001, p.507).

Beyer and her colleagues have examined how the opinions of Management Information Systems and Computer Science students changed over a semester at an American university (Beyer, DeKuester, Rynes, & Deheer, 2004; Beyer, DeKuester, Walter, Colar, & Holcomb, 2005). They note the importance of not assuming that students' attitudes are fixed, as changes that occur during the course may remove the need for intervention. The results are mixed. For example, over time female Computer Science major students became more negative about the statement that Computer Scientists enjoy being around people whereas male Computer Science major students became more positive about this statement.

The prior research that has considered changes in attitudes generally has found that students become more negative, although when considering differences between genders as in the research of Beyer et al. (2005) a more complex picture emerges.

2.2 Content and understanding of Computer Science

Other research has considered what students or potential students know about Computer Science and what they expect from Computer Science courses. Greening (1998) surveyed Australian school children who were taking Computer Studies in the last two years of high school. Of the sample, 58% were unable to give a definition of Computer Science and the remainder gave partial definitions. Even students who were planning to study computing at university did not have a good understanding of the discipline. The students were also unclear about what they would expect to learn in an introductory computing course with answers ranging from programming, general usage skills, use of applications, and hardware to a repeat of the high school syllabus, and 39% gave null or trivial answers. When asked about whether mathematical ability was necessary to write programs, 46.4% were neutral, 35.8% agreed and the remainder disagreed. Of the sample, 72.2% thought all first computer courses should teach spreadsheets, 71.8% database design, 70.3% graphics programming, 67.8% multimedia, 61.9% programming, 58.3% network programming, 48.9% algorithm design, 47.6% games programming, 31% integrated circuit design, and 25% artificial intelligence. The results indicate that students appear to have little understanding of the nature of Computer Science or what to expect in an introductory university course. Scragg and Smith (1998) found that students in an introductory computing course at an American university believe that mathematics plays a large role in Computer Science.

Moore, Wick, & Peden (1994) developed a tool to investigate attitudes. They considered enjoyment of Computer Science, the value of Computer Science, the role of programming in Computer Science and the role of science in Computer Science. Of interest to the research reported in this paper, is the fact that there was high agreement with the statement that 'There is more to Computer Science than programming'.

Australian Bachelor of Computing and Information Technology students found their courses different to what they had expected: "the course was regarded as very hard, involving more programming and maths than was expected" as well as "too theoretical and impractical" (Nielsen et al., 1998, p.89).

In South Africa, research investigating students starting Science degrees found that students were unclear of the content of Computer Science with many associating it strongly with programming, some with the use of application software such as word processors and others only giving nonspecific answers (Herbert, 2000). Among students planning to study Computer Science about a quarter of the comments were not specific, and half the sample mentioned programming. The predominant reason given among students who had chosen not to study Computer Science was a desire to follow a different field of study; other reasons mentioned included Maths marks that were too low for admission, lack of interest in computers and computing careers and lack of knowledge about computers and Computer Science (Herbert, 2000).

In general, students did not have a clear idea of the nature of computing or Computer Science, as is shown by responses in surveys from a number of countries. This may be explained in part by the complexity of defining what Computer Science is (Denning et al., 1989). Other research has investigated perceptions of careers, and this is discussed in the following section.

2.3 Perceptions of computing careers

As well as not understanding what will be taught in Computer Science courses, students are confused about careers. Research by Durndell and Thomson (1997) showed that students studying business or natural science at a Scottish university choose not to study computing because of misperceptions about the careers available. They did not want to work in front of a computer all day and believed studying computing would remove them from contact with people. They also expressed greater interest in people than objects. Other research has shown similar results about the solitariness of computing careers (Greening, 1998; Clarke and Teague, 1996; Symonds, 2000). Another misperception, particularly amongst female respondents, is that studying computing leads to secretarial and administrative careers (Craig, 1997; Clarke and Teague, 1996; Symonds, 2000).

Nielsen et al. (1998) found that the Australian students in their sample were unclear about the differences between using computer systems and developing software for them. In contrast, the American and Finnish students had a better understanding which may be explained by the fact that they were, on average, older. They also found that students had difficulty in describing the types of career they wanted, although this was less of a problem for the more mature students and the Asian students.

Research has shown that students perceive that computer professionals will use a computer in their work, design computer programs, undertake system analysis and design (Craig, Paradis, & Turner, 2002). The sample consisted of students from Australia, Britain, Hong Kong and the USA taking first year introductory programming courses who selected activities from a given list. The same questionnaire given to second-year Computer Science students in South Africa showed that the students perceive that computer professionals undertake system analysis and design, design computer programs and create databases (Craig, Galpin, Paradis, & Turner, 2002). Few students chose the following activities: computer games, writing documentation, creating spreadsheets and doing mainly word processing.

Another South African study of students starting Science degrees found that approximately 40% of the sample did not know what jobs a Computer Science degree would qualify one for; considering only the students planning to study Computer Science, the proportion of those not knowing was about a third (Herbert, 2000). Around two-thirds of the sample believed that there were good jobs for Computer Scientists (Herbert, 2000).

2.4 Gender differences in attitudes

Much has been written about gender differences in computer usage and attitudes. Surveys of this research as well as suggested explanations for the differences can be found in the papers of Teague (1996) and Gürer and Camp (2002). Meta-analyses have been conducted for existing studies. For example, the meta-analysis done by Liao (2000) considers computer attitudes which showed slight gender differences exist. The meta-analysis of Whitley (1997) showed that males gender-stereotype computers more, have greater self-efficacy and more positive attitudes. Whitley's meta-analysis found no difference in computer usage or computer beliefs. Due to the existence of survey articles and meta-analyses, and for space reasons, this section will only focus on South African research.

Moore investigated secondary school students' attitudes towards computers using a sample of Matric students (Moore, 1994) (the Matric school leaving qualification is explained in Section 2.6). She found female participants were more negative towards computers. However, computer usage was more correlated with positive perceptions than gender was; the boys were more positive because they used computers more often. Pike, Hofer, & Erlank (1993) surveyed English-speaking Western Cape Matric students and found that gender did not affect attitudes towards computers. In contrast, a study of Afrikaans-speaking secondary school students showed that the female students were less interested in Computer Science (Swanepoel, 1990).

At the University of the Witwatersrand, first year male Computer Science students were more confident about their ability to study computing than their female peers (Sanders and Galpin, 1994). They were also more likely to have chosen to study Computer Science because of their liking for computer games. The female students were more positive about women's ability to succeed at Computer Science. A subsequent study (Herbert, 2000) showed that amongst male and female students who were planning to study Computer Science, almost a third of male students were vague about the content of Computer Science whereas few women were vague, and about a fifth of the female sample expected programming and how to use a computer as the course content with few men giving this response.

Clarke and Finnie's research found no significant gender differences between Commerce students in their attitudes to computers (Clarke and Finnie, 1998) in contrast to Finnie's earlier study in which female students were less positive, had a greater fear of computers and less technical appreciation (Finnie, 1987). A later survey considered Science, Commerce and Arts students at the same university and found that males were more positive about computers, but found no significant difference in terms of anxiety (Clarke, 2000). Smith and Oosthuizen (2003) found that female students were more apprehensive about the impact of computers on society.

In summary, prior research into gender and attitudes in South Africa has not given consistent results, with differences found in some cases and not others. Because of the prevalence of negative attitudes and misperceptions, academics have considered how introductory computer science should be taught. The next section reviews different approaches to introductory courses.

2.5 Approaches to introductory Computer Science courses

Computing Curriculum 2001 (The Joint Task Force on Computing Curricula, 2001) highlights the different ways that introductory Computer Science courses can be taught. Historically, introductory courses have focussed on programming, and this remains the predominant model in the USA, and probably worldwide. There are a number of weaknesses with this approach including providing the impression that Computer Science is programming, delaying foundational theoretical issues until later in the curriculum, focussing on the syntax of a particular language without dealing with algorithms and disadvantaging students who have no programming experience without challenging those with experience (and possibly bad programming habits). Programming-first curricula are classified as imperative-first, objects-first and functional-first.

Computing Curriculum 2001 also presents three non-programming introductory curricula: breadth-first which aims to provide an overview of the field of Computer Science, algorithms-first which focusses on algorithms without programming and hardware-first which starts with circuits and considers successive layers of abstraction. Another dimension of this debate is whether discrete mathematics is given separately or as part of the introductory Computer Science courses. The first year courses taught at the University of the Witwatersrand (which will be described in Section 2.7) can be classified as breadth-first courses.

As this document considers the teaching of Computer Science in South Africa, some background on the educational system is presented next.

2.6 Education in South Africa

The South African education system requires all learners to complete seven years of compulsory primary schooling starting from the year in which the learner turns seven (Grades 1 to 7) followed by up to five years of secondary schooling (Grades 8 to 12). A learner can leave school at the end of the year in which they turn fifteen or complete Grade 9. In their secondary education, students typically do ten subjects in Grades 8 and 9 and this number is then reduced to six from Grade 10 when the learners are typically 14 to 15 years old. These six subjects must include at least two of South Africa's eleven official languages. Subjects can be taken at either the Higher Grade or Standard Grade levels. The final school leaving examination is the Senior Certificate Examination which also serves as the matriculation exemption examination (Matric) and is based on the subjects taken in Grades 10 to 12.

The results of the Matric exam are used to determine whether the learners will be offered entrance into university and in many cases used to determine if they will be allowed to register for a particular degree. In order to be accepted to study Computer Science, a sound overall Matric symbol and good pass in Mathematics is usually required. A standard Bachelor's degree takes a minimum of three years of full-time study (Engineering and Medicine are different), after which students can read for a Honours degree (one year of full time study). Some students then elect to complete Higher Degrees – Masters or PhDs. Most graduates with majors in Computer Science will initially have obtained Bachelor of Science degrees, and those students majoring in more business-related areas such as Information Systems obtain Bachelor of Commerce or similar degrees.

The South African education system is still affected by the legacy of the apartheid system. Students in poorer areas such as rural areas and the formerly black urban townships are still educationally disadvantaged because of their schools' inability to levy fees to increase resources, particularly computer laboratories (Vally and Dalamba, 1999). This means that many students do not gain exposure to computers at school and do not have the opportunity to select Computer Studies as one of their Matric subjects. In addition, the country suffers from a shortage of qualified mathematics and science teachers (Department of Education, 2001) and hence many students are not well enough prepared for the Matric exams and thus for entrance into university.

Our philosophy in the School of Computer Science is to provide our students with an education rather than training in specific aspects of computing. This means that our courses are structured to emphasise fundamentals and principles and concentrate less on specifics of particular programming languages or operating systems etc. The effect of this is that our courses are more theoretical and abstract than many of the computer science courses at other South African universities. In addition, we see Computer Science as a science subject which is strongly allied to mathematics. This is reflected by our position as part of the Mathematical Sciences grouping in the Science Faculty at Wits. We expect our students to have a strong grasp of Mathematics and to appreciate the rigour of the discipline.

Our curriculum changes over the years reflect this stance (Mueller, Rock and Sanders, 1993; Sanders and Mueller, 2000). Our current first year curriculum, introduced in 1999 (Sanders and Mueller, 2000), is intended to stress fundamentals and principles of the discipline and to emphasise the mathematical nature of Computer Science. The aims of the course are to give an introduction to Computer Science; to cover the basic principles upon which the courses in subsequent years build; to develop essential skills required in subsequent years; to motivate the students in the discipline of Computer Science; to develop the correct attitudes required for tackling problems in the discipline; and to break down the "Computer Science is programming" attitude which was prevalent among students.

In addition, the curriculum was designed to make the first year courses accessible to all students with the required mathematics ability/potential, and to shrink the gap between the various groups in the student population. By emphasising the mathematical aspects of Computer Science, and not focussing on programming, students without computer experience before university but with mathematical ability can succeed in our courses. This means that students who are disadvantaged by lack of facilities at school still have access to Computer Science at Wits.

At Wits the academic year is divided into 4 blocks each comprising approximately 7 weeks of lectures. In Computer Science, a course is typically taught over one block or in parallel with another course over two blocks at "half pace". At first year level Computer Science students take four courses. Additionally, there is a component associated with each course called Basics which is aimed at developing computer and programming skills. There are five lectures (45 minutes each), a tutorial (also 45 minutes) and a two hour laboratory session each week. The courses in the new first year curriculum are discussed below.

- **Blocks 1 and 2** – Basic Computer Organisation (BCO) and Fundamental Algorithmic Concepts (FAC) are taught in parallel.
  BCO – Propositional logic, boolean algebra, relationship between logic and hardware, introduction into basic hardware building blocks, automata, simple von Neumann model, low-level programs (2 lectures per week)
  FAC – Basic graph theory, proof techniques and simple proofs (direct, inductive, contradiction, constructive), formal specifications, study of well known algorithms, verification of simple algorithms, analysis of simple algorithms, overview of analysis of algorithms, overview of programming languages and compilers (2 lectures per week)
  Basics – Basic computer hardware, Linux, email, syntax of the chosen programming language, translating an algorithm into code, desk checking, compiling, error detection, error correction (1 lecture per week)
- **Block 3** – Data and Data Structures (DDS)
  DDS – Representation of data, data structures (such as lists, stacks, queues), recursion, dynamic data structures, study of well known algorithms on these data structures, analysis of these algorithms (4 lectures per week)
  Basics – Plenty of exercises using arrays, procedures, records, pointers and files (1 lecture per week)
- **Block 4** – Limits of Computation (LOC)
  LOC – Theory of computation (including undecidability and the halting problem), social issues (including ethics and responsibilities of scientists and professionals), artificial intelligence (including game-playing, expert systems and natural language processing) (4 lectures per week)
  Basics – Overview of databases, overview of networks, the use of the internet, markup languages (1 lecture per week)

We believe that these courses give students the knowledge and skills to succeed in our later courses.

The research presented in this document considers how our courses affect our students' perceptions of the discipline. The next section describes the methodology used after which the results of the research are presented and discussed.

3 Methodology

3.1 Sample and questionnaire

The sample consisted of 46 students, all of whom were attending university for the first time. They were all registered for our Computer Science I courses. There were 15 female students and 31 male students, with ages ranging from 17 to 21 with 78.2% of the sample 17 or 18 years old. The students were surveyed in the years 2000 and 2002, and there were no significant differences found using the Fisher Exact Test (Sheskin, 2000) between the two year groups in terms of sex, age, Matric maths mark, type of school (co-ed or single-sex), race, Computer Studies in Matric, computer usage, home computer and plans to major in Computer Science. Demographic information is described in Table 1.

Since our first year consists of 4 courses spread over the year, and because some courses are prerequisites for other courses, the students for whom we obtained two questionnaires represent students who had success in the first two courses, and had done well enough in the mid-year exams, and hence were representative of those who are likely to persist to second and third year.

As we wanted to investigate the perceptions of new students, we excluded from the sample any students repeating the course or any student who had been at any university prior to the year of the survey.

The students completed a questionnaire on the first day of the course (which was also their first lecture at university) and then later in the year completed a similar questionnaire which had some additional questions relating to the course. The questionnaire appears in the fuller technical report (Galpin and Sanders, 2005). Questionnaires were matched before analysis.

3.2 Analysis

As our aim was to investigate how student perceptions changed, we used McNemar's test and the binomial sign test (Sheskin, 2000) to determine whether results showed significant changes. Both of these tests assess whether there has been a significant change in a specific direction. For example, in a dichotomous question (one that requires students to answer yes or no), the McNemar test gives a significant result when the proportion of students that answered yes in the first questionnaire and no in the second questionnaire, is significantly larger than the proportion of students who answered no in the first questionnaire and yes in the second questionnaire (or vice versa), where the proportion is calculated with respect to the number of students who changed from the first questionnaire to the second. The null hypothesis is that the proportions are equal, that is they are both equal to 0.5. The students that gave yes in both questionnaires or no in both questionnaires are not considered in the test. The binomial sign test is similar but applied to ordinal data (in this case, an Agree, Neutral, Disagree scale). As in the McNemar test, students who answered the same in both questionnaires are not considered. The binomial sign test determines if the proportion of respondents who have moved towards the Agree end of the scale is significantly larger than that of those who have moved towards the Disagree end of the scale (or vice versa). The null hypothesis is the same as that of the McNemar test.

4 Results

This section presents the results of the research, and starts by considering the closed questions, then open-ended questions about jobs and expected course content, and finally an assessment of the courses. The results from the closed questions show that in general, student attitudes and perceptions appear to be resistant to change. These are presented in Tables 2 to 5.

For Tables 2 to 4, the first four columns of these tables give the percentage of students with the combination of answers indicated in the column heading. This percentage is calculated with respect to the value given in the n column. This value is the number of students that answered some combination of yes and no on the two questionnaires – students that did not respond on one of the questionnaires are excluded. The next column gives the probability that there is a significant difference in the proportion of students switching, and an asterisk next to this value indicates whether it is significant at the 0.5 level, that is this indicates that the probability that the differences observed are due to chance is less than 5%.

4.1 Content of Computer Science and students' interests

Table 2 presents the data for a number of statements with Yes and No responses. It covers questions relating to the content of Computer Science and students' interests. As can be seen, significant results are only found for the first two statements, one asking about their understanding of their knowledge of Computer Science and the other asking about their perception of the relationship between Computer Science and mathematics. These two statements will be discussed in more detail below.

For the rest of the statements, a large majority of students responded yes on both questionnaires, and there were few students who changed their opinion between the two questionnaires. For the statement about studying Computer Science to learn programming, there were fewer double yes responses and more change in opinion, although almost equally balanced in terms of direction, hence the lack of significant difference. Students were positive about their ability to use new technology and the availability of good jobs.

Further analysis of the data was done for all questions in Table 2. First, those questions for which there were no significant differences for the whole sample, were analysed by Matric Computer Studies, computer usage, and encouragement to study a science degree. No significant differences were found for any of these variables.

Further analysis was also done for the two questions in Table 2 where there were significant differences, and these are discussed in the next section. Note that there is a significant correlation (Φ co-efficient 0.9014 (Sheskin, 2000)) between Matric Computer Studies and computer usage: 100% of those who had done Computer Studies had programmed, and of those who had programmed only 10% had not done Computer Studies. Hence, we have chosen only to indicate the responses for computer usage.

4.1.1 Understanding of Computer Science

At the start of the academic year, only 55.8% of the sample (who answered the question) were convinced that they had a clear idea of what Computer Science involves. By the end of the year, this had increased to 81.4% with 27.9% switching from no at the start to yes later. The proportion switching was found to be significant using the McNemar test. Further analysis was done, and as is shown in Table 3, significant differences were found for female students, those who had not taken Computer Studies for Matric, those whose prior computer usage included applications, but not programming, those who had been encouraged to take a science degree, and those who had a computer at home. Note that for female students, those who did not take Computer Studies for Matric and those whose prior computer usage was applications, more than 60% of the respondents answered no in the first questionnaire.

4.1.2 Relationship between Computer Science and mathematics

Of the sample, 24.4% started the year with the belief that Computer Science and mathematics are closely related, but later in the year had changed their opinion. None of the sample changed their opinions from no to yes. This gave a significant result. Further analysis found significant differences, and as shown in Table 4, these occurred for male students, students who had taken Computer Studies for Matric, those who had done programming before university, and those not encouraged to study a science degree.

4.1.3 Content statements

For this part of the questionnaire, students were asked to indicate whether they agreed, disagreed or were neutral with respect to a list of statements. The results are shown in Table 5. The first three columns of figures indicate the percentage of students who did not change their opinions, the fourth column those who changed from Disagree to Neutral or Agree, or from Neutral to Agree, and the fifth column those who changed from Agree to Neutral or Disagree, or from Neutral to Disagree. These percentages are calculated with respect to the value in the n column which represents the number of students who answered Agree, Neutral or Disagree on both questionnaires and excludes the students who did not respond or chose Don't Know on one of the questionnaires. The column headed p represented the probability calculated by the binomial sign test, and significant results are indicated by an asterisk. Because a two-tailed test is required, the value used is 0.025 (Sheskin, 2000).

As can be seen from Table 5, significant differences were only found for two statements. For both statements, a significant portion of students moved towards agreeing with the statements, one which stated that working with computers is boring, and one which stated that Computer Science is not interesting because it is about machines rather than people. About 20% of the sample became more negative about working with computers and Computer Science, with around 70% remaining positive (reflected by their disagreement with the statements). Most of the movement was from disagreement to neutral. For the first statement, of the 9 students who changed their opinion, 8 moved to neutral, and for the second statement, of the ten students who changed their opinion, 70% moved to neutral. There were no significant changes by gender.

For the rest of the statements, the students were mostly positive about employment prospects, fairly definite that Computer Science was not mainly about word processing, and had mixed opinions about whether Computer Science involves mainly programming with changes of opinion in both directions. Students also had mixed opinions about whether they wanted to learn about using the computer to solve problems, or whether they wanted to learn about the computer itself, with changes in opinion in both directions. There were no significant changes by gender.

4.2 Open-ended questions

Students were asked three questions: what jobs a Computer Science degree would enable one to do, what they personally wanted to do with their degree and what they expected to learn in a Computer Science course. Table 6 gives possible jobs that could be done with a Computer Science degree as well as personal goals. Table 7 gives a gender breakdown of the first two questions. Table 8 presents what students expect to learn in a Computer Science course, with percentages for the whole sample, as well as a breakdown by gender. The job descriptions and course content items that appear in the tables represent those where 10% or more of the sample mentioned in it in at least one of the three questions.

Since prior research has shown that students often assume Computer Science is programming or are unclear about what Computer Science is, the discussion of the tables that follows will concentrate on these aspects. Details of other responses can be found in the tables.

When asked in the first questionnaire about what jobs they would be qualified to do with a Computer Science degree (first column in Table 6), 21.7% didn't know, 67.4% said programmer and 19.6% of respondents mentioned jobs for which it is very unlikely that a Computer Science degree would be required (these will be referred to as non-degree jobs in the rest of this section). Similar levels of response were given in the second questionnaire (second column in Table 6), with the percentage stating programmer increasing slightly. In both questionnaires, 17% mentioned programmer only, and 54% said programming on both questionnaires.

Students were also asked what they would like to do once they completed a Computer Science degree. For the first questionnaire (third column in Table 6), 37% did not know, 30.4% mentioned programming and only 4.3% of respondents mentioned non-degree jobs. In the second questionnaire (fourth column in Table 6), the level of response had dropped for those that didn't know, and for programming, but there was an increase in those mentioning non-degree jobs. On the first questionnaire, 20% gave programmer only, and on the second, 26% gave programmer only. 17% gave programming both before and after.

Students were also asked about what they would expect to learn in a Computer Science course (Table 8). 13% said they didn't know or gave no response in the first questionnaire compared with 23.9% in the second questionnaire. Half the sample said programming on the first questionnaire and half on the second, but only 26% said programming on both. For the first questionnaire, 9% said programming only, and on the second, 15% said programming only.

Looking at those who said that they had a clear idea of Computer Science and those who did not, the number in each group that said programming was about 50% on both questionnaires. Since the number that were not clear about Computer Science on the second questionnaire was small (only 8), it is not possible to do further analysis of the differences.

4.2.1 Gender

When considering the responses of male and female students to the questions about jobs separately (Table 7), no strong pattern emerges. Fewer female students chose 'Don't know' or gave no response in the second questionnaire

(33.3% compared with 13.3% for the question about jobs) than on the first questionnaire whereas more male students chose ‘Don’t know’ or gave no response on the second questionnaire (16.1% compared with 22.6% for the question about jobs). A similar pattern occurred when asking the students about what they wanted to do with their degrees.

When asked about possible jobs (first and second columns of Table 7), of the female sample, 53.3% mentioned programming in the first questionnaire, and 66.7% in the second, showing an increase. 40% mentioned programming both before and after, indicating that it was not necessarily the same students on both questionnaires. When asked about their own career plans (third and fourth column of Table 7), 20% of the female sample mentioned programming on the first questionnaire, and 13.3% on the second, showing a decrease. 7% mentioned programming both before and after.

In the male sample, 74.2% mentioned programming on the first questionnaire and the same percentage on the second questionnaire. However, only 61% mentioned programming on both questionnaires. Considering their own career plans, 35.5% of the male sample mentioned programming in the first questionnaire, and 32.3% in the second, showing a slight decrease. 16% mentioned programming both before and after.

For both the male and female students, there was an increase in those who didn’t know or gave no response when asked about the content of a Computer Science course (Table 8). For female students, the percentages were 26.7% on the first questionnaire and 40% on the second, and for male students, 6.5% and 16.1%.

For female students, when asked about course content, 60% said programming on the first questionnaire and 53.3% on the second, compared to 45.2% for male students on the first, and 48.4% on the second. 26.7% of female students said programming both before and after, compared with 25.8% of male students.

4.3 Feedback on the course

In the second questionnaire, students were asked questions about the first year courses. The results are given in Table 9. The majority of the students found the course different to their expectations. When asked an open-ended question about how the course differed from their expectations, 39.1% of the total sample indicated that the course was more theoretical (7 responses), mathematical (1 response) or had more fundamental content (3 responses) than expected; or had less practical (6 responses) or programming (3 responses) content than expected. Only one person commented that the course was less mathematical than expected.

Of the sample, 17.4% thought the courses were unenjoyable, 8.7% uninteresting, 4.4% not challenging, and 8.7% not preparation for a career. About a quarter (23.9%) thought the courses were not useful for other courses, which is reasonable as we focus on teaching Computer Science.

In terms of their learning experience, few thought they didn't learn a lot of facts, and only 8.7% felt that they didn't improve their ability to do independent study. Note, however, that for each of these questions, a moderate proportion of the sample chose Neutral as an option, so are not prepared to be definite in either direction.

Further analysis was done by gender, whether the course was different from expectations, and whether Computer Studies was taken in Matric using the Fisher Exact Test (Sheskin, 2000). No significant differences were found in the responses of the different groups.

5 Discussion

In general, students' opinions and perceptions show little change, indicating that these are fixed and not influenced by the courses. Even in cases where there is a high percentage of change in opinion, there is often change in both directions (for example, when asked about studying Computer Science to learn programming or whether Computer Science is mainly about programming).

A limitation of this research is the fact that this sample represents those who succeeded at the mid-year exams. This may mean that we excluded from the sample the students who may have arrived with the least knowledge of Computer Science, and for whom the course would have been of most benefit. This is a problem we need to consider for further research.

5.1 Understanding of Computer Science and computing careers

There was a significant change in students' own assessment of their understanding of Computer Science, and with further analysis, this change was shown to be significant amongst women, those who did not study Computer Studies in Matric, those who had not programmed and those who were encouraged to study a science degree. For all of these groups, their prior assessment of their understanding was low, and this change represents a real shift in their assessment of their knowledge.

However, when considering their actual knowledge, half the sample said programming on the first questionnaire and half the sample said programming on the second questionnaire, and there were no differences between those who believed that they had an understanding of Computer Science and those who did not. Looking at the responses given by more than 10% of the sample, algorithms was mentioned in the second survey and not in the first, and software, how computers work and how to use a computer, occurred in the first survey at the 10% level, but not the second. This represents some shift in understanding, although many students still see programming as the thing they will learn in Computer Science, suggesting that this perception has not been affected by the course. Two facts mitigate against this interpretation: first, only a quarter of the sample gave programming before and after, suggesting shifts in both directions; and second, only 9% of the sample in the first questionnaire and 15% in the second questionnaire only mentioned programming, indicating that in general, the perception is that programming is part of Computer Science, rather than all of Computer Science. This suggests that students have a broad understanding of Computer Science, or at least broader than just programming (though unfortunately not due to the course).

Assessing the students' understanding of the jobs available for Computer Science graduates, shows that again programming is predominant, at around 70% in both questionnaires, but with only 17% in both questionnaires giving only programming, again showing that there is some breadth of knowledge. Around 20% of the sample chose jobs that are very unlikely to require a Computer Science degree, and worryingly, in the second questionnaire, 13% of the sample indicated that a Computer Science degree qualified one for the job of technician.

In terms of their own job prospects, programming was at around 30%, and about a third did not know, indicating that many students are unsure about their future direction.

An interesting fact that came to light when assessing students' perceptions of careers is that there are very few job titles for which one can state definitely that a Computer Science degree is required: researcher is one, but for many other jobs there are multiple routes to these positions, for example, programmer, system administration, networking. This diversity of routes may lead to the confusion experienced by students.

The results from our research are similar to those of Greening (1998), Nielsen et al. (1998) and Herbert (2000) where respondents appeared to have little understanding of the nature of Computer Science or the types of careers that are available.

Questions relating to the availability of good jobs which appear in Table 2 and Table 5 show that students are positive about their prospects even if they do not know what they are. This agrees with the results of Herbert (2000).

Both male and female students became more uncertain about the content of Computer Science courses (or perhaps they felt there are so many aspects to Computer Science that they were daunted in their attempt to write down their understanding). Women were more uncertain on both questionnaires than men, which is interesting in light of the significant change in their own assessment of their understanding. There was slight decrease in the number of women who said programming and a slight increase in the number of men, with more women than men mentioning programming.

Female students became aware of more jobs during the year, particularly programming. They also became more specific in terms of their own career goals – on the first questionnaire half the sample didn't respond, compared to a quarter on the second questionnaire. They also showed an increase in their awareness of careers for which a degree is unlikely to be required, specifically web-related jobs and technician.

Male students appeared to have less knowledge later in the year – on the second questionnaire, almost a quarter did not respond when asked about what jobs could be done with a degree, and almost a third did not respond when asked about personal goals.

Comparing these results to those of Herbert (2000) is interesting. Far more students said they didn't know or didn't respond in this research, and far fewer gave vague answers – in fact, most answers were specific.

5.2 Relationship between Computer Science and mathematics

Although the FAC course deals with mathematical proofs relating to graph theory, as well as a mathematical model of a program, the BCO course deals with logic and the LOC course deals with theory of computation, a significant proportion of students changed their minds about how closely mathematics and Computer Science are related. This is clearly an issue for concern, as we do believe there are strong links between the two disciplines and choose to teach material in first year to highlight that fact. This result contradicts the finding by Scragg and Smith (1998) that students believe that mathematics plays a large role in Computer Science.

A possible explanation is that the other first year mathematical science courses that the majority of students do (Mathematics I Major, and Computational and Applied Mathematics I) focus on continuous mathematics and that the students do not see the link between these subjects and the more discrete mathematics shown in the Computer Science courses. Without understanding why students give the answers they have, it is not possible to reach a definite answer, and this is an area for further research.

Further analysis showed that significant differences were found for male students, those who had done Computer Studies for Matric and those who had programmed before, and those not encouraged to do a science degree. Anecdotal information has identified male students who have programmed before as those who are most likely to assume they know the course material already even though most often they only know how to program and know very little Computer Science. It would have been our expectation that this group would arrive with the idea that Computer Science and mathematics are separate disciplines and finish with an understanding of the links. Why the opposite is the case could possibly be explained in terms of the continuous/discrete split, if much of their prior programming experience had involved continuous mathematics. This requires further research.

5.3 Working with computers

In Table 5, the only two statements with significant differences in changed proportions deal with working with computers, and whether Computer Science is interesting. In both cases, around 20% of the sample became more in agreement that working with computers is boring or not interesting. Most of the changes of opinion are from positive to neutral as opposed to a shift to negative. Although this is somewhat disappointing, it must be borne in mind that we do not have any data to compare this with, so it may be the case that a more traditional first year Computer Science course focussed around programming, would have led to more negative responses. Additionally, many of the students (67.4%) found the course challenging, and it is our experience that students can be much more negative about challenging courses.

The change to a more negative attitude is consistent with the research of Finnie (1987), Burger and Blignaut (2004) and Staehr et al. (2001) where attitudes were more negative after courses.

In the research of Beyer et al. (2005), it was found that on average, male Computer Science major students became more positive about computers being fun over time, whereas female Computer Science majors became more negative. In the research presented here, the trend was for both male and female students to move away from disagreeing with statements about computers being boring or not interesting, so both male and female students became more negative.

5.4 Perceptions of the course

The majority of the students found the course different to their expectations, and 30% of the total sample indicated that the course had more theoretical or mathematical or less practical content than expected. Relating this back to the closeness of mathematics and Computer Science, perhaps students distinguish between theoretical and mathematical as concepts. This is supported by the fact that 7 students said the course was too theoretical compared to 1 that said the course was too mathematical. This differs slightly from the sample of Nielsen et al. (1998) which found their course different, but emphasised that it had both more programming and theoretical material than expected.

In light of the high number of students (around two-thirds) who found the course challenging, the fact that there were only 8 students who found the course unenjoyable is a positive sign. About a quarter thought the courses were not useful for other courses, which is not unexpected as our focus is teaching Computer Science.

5.5 Summary

In summary, the courses were not what students expected, however, they did not appear to shift students' perceptions appreciably. Students did become more positive about their own assessment of their knowledge of Computer Science. In terms of their actual knowledge, a level of breadth was shown, but programming was predominant as a job option and as expected course content. Since the course does not focus on programming, this could lead to students finding the course differing from their expectations. Although a large proportion felt the course was more theoretical than expected, there was significant shift away from the perception that mathematics and Computer Science are closely related.

The question arises about what has more influence on students – what their lecturers say in class, or what the students are required to do in tutorials, laboratories and preparing for exams, or factors outside the courses that are beyond our control. Perhaps the fact that they spend at least 2 hours a week in front of a computer in a laboratory and have assignments based on programming has far more influence than what lecturers say about the breadth of the discipline.

There are some indications that there may be two different groups in the class: those who expect a lot of programming, and then find that there is less than they expect, and a group that expects little programming and finds that there is more than they expect, and further research may uncover this.

Clearly these results give us reason to reflect on what we are doing, and how we can change the course to achieve our aims and objectives. It seems possible that any Computer Science course that deals with some programming as our first year courses do, will emphasise the link between the two, but we need to further balance that with other material in the courses. There are a number of examples of breadth-first courses in the literature such as the courses discussed by Phillips, Stevenson, & Wick (2003), Burch and Ziegler (2004), Shannon (2003) and Powers (2003). The non-programming course offered by Turner and Turner (2005) which focussed on problem-solving in Computer Science is a case in point. This course helped retain students by allowing students to find out if they were likely to succeed in later Computer Science courses, helped with the fact that much of the exposure in high school to computers has little to do with Computer Science, and showed students that there was more to Computer Science than programming. It also helped with the retention of female students.

5.6 Curriculum and pedagogy

In the short term we will not be able to make significant changes to our curriculum (governmental and University procedures need to be followed) but there are some changes which we can make to try to address some of the disappointing results from this study. The first such change would be in reordering the topics in our first year curriculum (see Section 2.7 for the detail of the topics and their current ordering). Currently we teach BCO and FAC in parallel in the first half of the year. Although both of these topics concentrate on Computer Science fundamentals, FAC (and to a lesser extent BCO) includes some programming which might still be sending the wrong message to our students. We have already made the decision to move the LOC topic to the beginning of the year. This topic is more theoretical and abstract than the other topics and we believe is more suitable for exposing the students to the true nature of the subject. The fact that the topic is more theoretical also means that students who have done computing and/or programming at school are at less of an advantage than in the current ordering. The modified version of the topic will concentrate more on mathematical models of computation and logic than the current version of LOC in an attempt to stress the link between mathematics and Computer Science. In addition, we will be integrating the Basics course more with the mainstream material. Instead of simply teaching skills and hoping the students will make the right connections, we will be introducing the various skills at the time they are needed and being explicit about where the skills we are teaching fit into the study of Computer Science. For example, instead of just teaching the students about the use of arrays in some programming language, we will discuss data structures and why they are important. We hope that these changes will make the link between mathematics and Computer Science more explicit and also further breakdown the Computer Science equals programming mentality.

In addition, to the major changes discussed above we can make smaller changes to the way that we present and discuss material in the first year course. To stress that Computer Science and Mathematics are strongly linked, we should be even more precise and rigorous about the use of language and mathematical notation. We should be explicit about how concepts like graphs, sets, logic, etc. belong in Mathematics and in Computer Science. We should make explicit links between concepts or techniques which are useful in both disciplines. To address the fact that programming is part of but not the whole of Computer Science we should be more explicit about what we believe the role of programming is in a Computer Science degree.

The fact that students became less positive about working with computers could have been due to the laboratory environment that we had at the time but it is clearly something that we should be concerned about. Our current environment is much better, with faster machines with a clean and simple operating system and better network access. We intend to use this improved environment to stress that computer science is a broad and challenging discipline. Using simulators for theoretical models of computing such as Turing machines and finite state machines, as well as essay writing about aspects or history of Computer Science are two possible examples of how we would use the laboratory time.

5.7 Further research

It appears that students may believe that there is more to Computer Science than programming, but they may not be able to articulate what these additional things are because of lack of knowledge. Can we develop a questionnaire that will help with this, perhaps by providing closed questions or a standard instrument for Computer Science such as the one developed by Moore et al. (1994)? An instrument has been piloted at Wits, but further work is still required (Mahomed, 2005). Even if we take this approach, there is still the issue of what people understand by terms such as 'mathematics'. A possible solution is to do interviews to determine what students mean by the various terms that are in use – this may lead to much richer data and improved understanding.

As mentioned earlier, a limitation of this research is the fact that it does not include those who do not pass the mid-year exams, and who may have the least knowledge of Computer Science. Future research into perception changes could include a questionnaire before these exams.

Additionally, there is some indication that there are two distinct groups in the data whose opinions change in opposite directions, leading to an unclear picture. Further analysis of the data already collected has not been able to identify these two groups.

Another aspect we did not try to assess is how much students' perceptions are influenced by what their lecturers say and how much they are influenced by what the students actually do. Although the students may be hearing in lectures about the breadth of Computer Science, the exercises they do in laboratories and their programming assignments (they have both programming and non-programming assignments) may influence their opinions more strongly. Devising a way in which to assess this influence is not a simple task, and there may be other factors that outweigh either of these influences.

6 Conclusion

The aims of this research were to gain an understanding of first year Computer Science students' perceptions of the discipline and what they would study in their courses and to see how these perceptions changed across their first year of study. We surveyed incoming students in two years and then surveyed the same students later on in the same year. Our innovative first year curriculum is designed to emphasise the fundamentals of the discipline, and de-emphasise programming, and we wanted to assess its impact.

The results of the survey indicate that for most questions students' opinions did not change strongly or if there were changes then these were in both directions. For a few questions there were significant differences between the earlier and later surveys. The students became more confident about their understanding of the nature of Computer Science; the students became less convinced about a link between Computer Science and mathematics; and students became less positive about working with computers or working in the field of Computer Science.

Unfortunately, although the students became more confident about their understanding of Computer Science, we feel that their knowledge of the nature of the discipline (as reflected by their answers about expected course content and career opportunities) did not actually show that they had developed an appreciation of the breadth of Computer Science. Another disappointing finding was that the students moved towards feeling that mathematics and Computer Science are not related. This might be because Computer Science's strongest link is to discrete mathematics and our students do mainly continuous mathematics in their first year mathematics courses. This is certainly an area for future research.

Students also became less positive about working with computers. We included this question in the research as prior research has shown that students can become negative after a course. Our data showed the same pattern. This is an area for concern as these students are likely to work with computers heavily in their future careers, and a negative attitude towards computers may lead to students dropping out of the course. Although prior research has revealed this trend, there is little explanation of it, and understanding why this happens is an important area for future research.

Looking more closely at some of the data, female students showed a significant change in their assessment of their own understanding of Computer Science. Male students showed a significant change in their perception of the lack of relationship between Computer Science and mathematics. Female students became more aware of the jobs that they could take up with a Computer Science degree, and male students appeared to become less aware of these.

As with all educational research of this nature, it is difficult to conclude direct causality between the courses and our observations. However, by considering the various influences on the students, we are able to make some reasonable interpretations of these results. This is the best we can do within our particular pragmatic and ethical constraints. We believe our observations and interpretations, together with the detailed description of our particular circumstances, will provide useful information for other researchers and educators about teaching first year Computer Science at university level.

This study has given us a better understanding of our students and how their perceptions change during their first year of studying Computer Science but it has also raised a number of questions which we plan to consider in future work and our future teaching.

In future research, we would like to develop a better instrument to assess student perceptions of the nature of Computer Science and a first attempt at this has been made (Mahomed, 2005). We also suspect that there are at least two different groups in the class with different expectations of the course, particularly of programming and further research could identify these two groups. We also do not know what influences student perceptions the most, but at this stage, we have not devised a method to assess this.

In terms of teaching, we have a number of ideas for emphasising the theoretical and mathematical aspects of Computer Science, including rearranging the order in which material is presented in first year. This approach would also benefit students without computer exposure at secondary level. Other changes are to be more explicit about when we are using mathematical concepts and to provide an improved laboratory environment for activities that illustrate the breadth of Computer Science.

Acknowledgements: Thanks to all the students who took part in the research and to Hlami Huhlwane and Herman Tshesane for data capture.

References

Beyer, S., DeKuester, M., Rynes, K., and DeHeer, J. (2004). The temporal stability of gender differences in MIS students. In *Proceedings of the Tenth Americas Conference on Information Systems*, pages 1197–1204, New York, USA.

Beyer, S., DeKeuster, M., Walter, K., Colar, M., and Holcomb, C. (2005). Changes in CS students' attitudes towards CS over time: an examination of gender differences. In *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education*, pages 392–396, St Louis, Missouri, USA.

Burch, C. and Ziegler, L. (2004). Science of computing suite (SOCS): resources for a breadth-first introduction. In *SIGCSE '04: Proceedings of the 35th SIGCSE technical symposium on Computer science education*, pages 437–441, Norfolk, Virginia, USA.

Burger, A. and Blignaut, P. (2004). A computer literacy course may initially be detrimental to students' attitudes towards computers. In *SAICSIT '04: Proceedings of the 2004 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, pages 10–14, Stellenbosch, South Africa.

Clarke, M. and Finnie, G. (1998). Changes in entry-level university students' attitudes to computers for 1985–1997. *South African Computer Journal*, 21:26–32.

Clarke, M. C. (2000). Technological experience and technophobia in South African university students. *South African Computer Journal*, 25:12–17.

Clarke, V. and Teague, J. (1996). Characterizations of computing careers: Students and professionals disagree. *Computers & Education*, 26(4):241–246.

Craig, A. (1997). Women and Business Computing: where to from here? *gates*, 4(1):13–19.

Craig, A., Galpin, V., Paradis, R., and Turner, E. (2002). Panel: What is computing? The perceptions of university computing students. In *Grace Hopper Celebration of Women in Computing*, Vancouver, Canada. `www.cs.wits.ac.za/~vashti/sem/CGPT02GHC.html`.

Craig, A., Paradis, R., and Turner, E. (2002). A gendered view of computer professionals: preliminary results of a survey. *ACM SIGCSE Bulletin*, 34(2):101–104.

Denning, P., Comer, D., Gries, D., Mulder, M., Tucker, A., Turner, A., and Young, P. (1989). Computing as a discipline. *Communications of the ACM*, 32(1):9–23.

Department of Education (2001). *National Strategy for Mathematics, Science and Technology Education in General and Further Education Training.* Department of Education, Pretoria, South Africa. `http://education.pwv.gov.za/DoE_Sites/Maths%20and%20Science/Final-doc.pdf`.

Durndell, A. and Thomson, K. (1997). Gender and computing: a decade of change. *Computers & Education*, 28(1):1–9.

Finnie, G. (1987). Novice attitude changes during a first course in computing: a case study. *Quæstiones informaticæ*, 5(2):56–62.

Galpin, V. and Sanders, I. (2005). *Perceptions of Computer Science among first year students at the University of the Witwatersrand*. Technical Report TR-Wits-CS-2005-0, School of Computer Science, University of the Witwatersrand. http://www.cs.wits.ac.za/~vashti/pubs/GS05a.html.

Greening, T. (1998). Computer Science: Through the eyes of potential students. In *Proceedings of the Third Australasian Conference on Computer Science Education*, pages 145–154, Brisbane, Queensland, Australia.

Gürer, D. and Camp, T. (2002). An ACM-W literature review on women in computing. *ACM SIGCSE Bulletin*, 34(2):121–127.

Herbert, T. (2000). *Women role models in Computer Science at the University of the Witwatersrand*. Masters Research Report, Technical Report TR-Wits-CS-2000-16, Department of Computer Science, University of the Witwatersrand.

Lee, R. (1970). Social attitudes and the computer revolution. *Public Opinion Quarterly*, 34:53–59.

Liao, Y. (2000). A meta-analysis of gender differences on attitudes toward computers for studies using Loyd and Gressards' CAS. In *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, pages 605–611. http://dl.aace.org/1313.

Mahomed, S. (2005). *Relating underlying ideas of Computer Science to student perceptions*. Honours Research Report, School of Computer Science, University of the Witwatersrand.

McMillan, J. and Schumacher, S. (2001). *Research in Education: a Conceptual Approach*. Addison Wesley Longman, New York, Fifth edition.

Moore, C. (1994). Attitudes towards computers, the influence of sex stereotypes, experience, ownership and mathematics. *Unisa Psychologia*, 21(2):20–27.

Moore, T., Wick, M., and Peden, B. (1994). Assessing student's critical thinking skills and attitudes toward computer science. *ACM SIGCSE Bulletin*, 26(1):263–267.

Mueller, C., Rock, S., and Sanders, I. (1993). An improved first year course taking into account third world students. In *SIGCSE '93: Proceedings of the 24th SIGCSE technical symposium on Computer science education*, pages 213–217, Indianapolis, Indiana, USA.

Nielsen, S., von Hellens, L., Greenhill, A., and Pringle, R. (1998). Conceptualising the influence of cultural and gender factors on students' perceptions of IT studies and careers. In *SIGCPR '98: Proceedings of the 1998 ACM SIGCPR Conference on Computer Personnel Research*, pages 86–95, Boston, Massachusetts, USA.

Phillips, A., Stevenson, D., and Wick, M. (2003). Implementing CC2001: a breadth-first introductory course for a just-in-time curriculum design. In *SIGCSE '03: Proceedings of the 34th SIGCSE technical symposium on Com-*

puter science education, pages 238–242, Reno, Navada, USA.

Pike, N., Hofer, A., and Erlank, S. (1993). Effect of gender and occupational aspirations on attitudes towards computers. *South African Journal of Education*, 13(1):25–29.

Powers, K. (2003). Breadth-also: a rationale and implementation. In *SIGCSE '03: Proceedings of the 34th SIGCSE technical symposium on Computer science education*, pages 243–247, Reno, Navada, USA.

Ryan, C. (1994). Raising girls' awareness of computing careers. In *Proceedings of the Fifth International Conference on Women, Work and Computerization*, pages 547–559, Manchester, UK.

Sanders, I. and Galpin, V. (1994). A survey of attitudes to computing at the University of the Witwatersrand. In Adam, A., Emms, J., Green, E., and Owens, J., editors, *IFIP Transactions A-57, Women, Work, and Computerization, Breaking Old Boundaries—Building New Forms*, pages 209–223. Elsevier Science.

Sanders, I. and Mueller, C. (2000). A fundamentals-based curriculum for first year computer science. In *SIGCSE '00: Proceedings of the 31st SIGCSE technical symposium on Computer science education*, pages 227–231, Austin, Texas, USA.

Scragg, G. and Smith, J. (1998). A study of barriers to women in undergraduate computer science. *ACM SIGCSE Bulletin*, 30(1):82–86.

Shannon, C. (2003). Another breadth-first approach to CS I using python. In *SIGCSE '03: Proceedings of the 34th SIGCSE technical symposium on Computer science education*, pages 248–251, Reno, Navada, USA.

Sheskin, D. (2000). *Handbook of Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, Boca Raton, Florida, Second edition.

Smith, E. and Oosthuizen, H. (2003). An investigation of the attitudes of entry-level university students towards computers. In Mende, J. and Sanders, I., editors, *Proceedings of SACLA '03*, Pilanesberg, South Africa.

Staehr, L., Martin, M., and Byrne, G. (2001). Computer Attitudes and Computing Career Perceptions of First Year Computing Students. In *Proceedings of Informing Science 2001 – Bridging Diverse Disciplines*, pages 502–509, Krakow, Poland. `http://proceedings.informingscience.org/IS2001Proceedings/pdf/staehrEBKcompu.pdf`.

Swanepoel, C. (1990). Meting van belangstelling in die natuurwetenskaplike studierigtig (Measurement of interest in the natural sciences field of study). *South African Journal of Education*, 10(1):75–78.

Symonds, J. (2000). Why IT Doesn't Appeal to Young Women. In Balka, E. and Smith, R., editors, *Women, Work and Computerization: Charting a Course to the Future*, IFIP: Volume 172, pages 70–77. Kluwer.

Teague, G. and Clarke, V. (1991). Fiction and Fact: Students' and Professionals' Perceptions of Women in Computer Science. In Eriksson, I. V., Kitchenham, B. A., and Tijdens, K. G., editors, *Women, Work and Computerization*, pages 377–390. North Holland.

Teague, J. (1996). A structured review of reasons for the underrepresentation of women in computing. In *Proceedings of the second Australasian conference on Computer science education*, pages 91–98, Melbourne, Australia.

The Joint Task Force on Computing Curricula (2001). Computing Curricula 2001. *Journal of Educational Resources in Computing*, 1(3):1.

Turner, E. and Turner, R. (2005). Teaching entering students to think like computer scientists. In *SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education*, pages 307–311, St. Louis, Missouri, USA.

Vally, S. and Dalamba, Y. (1999). *Racism, "Racial Integration" and Desegregation in South African Public Secondary Schools*. South African Human Rights Commission.
http://www.sahrc.org.za/final_combined_school_racism_report.PDF.

Whitley, Jr., B. E. (1997). Gender Differences in Computer-Related Attitudes and Behaviour: A Meta-Analysis. *Computers in Human Behaviour*, 13(1):1–22.

			%			%			%
Age:	17	12	26.1	18	24	52.2	19+	8	17.4
Matric Maths:	HG A	14	30.4	HG B	17	37.0	Other	15	32.6
Race:	Black	12	26.1	Indian	14	30.4	White	16	34.8
Usage:	Prog	28	60.9	Appl	14	30.4	Never	4	8.7
Gender:				Female	15	32.6	Male	31	67.4
School:				Same sex	7	15.2	Co-ed	38	82.6
Computer Studies in Matric:				Yes	26	56.5	No	20	43.5
Home computer:				Yes	35	76.1	No	11	23.9
Plan to major in CS:				Yes	38	82.6	No	7	15.2

Table 1
Demographic information

	Unchanged		*Changed*				
Before	*Yes*	*No*	*No*	*Yes*			
After	*Yes*	*No*	*Yes*	*No*	*n*	*p*	
Do you have a clear idea of what Computer Science involves?	53.5	16.3	27.9	2.3	43	<0.01	*
Do you think Computer Science and mathematics are closely related?	68.9	6.7	0.0	24.4	45	<0.01	*
Are you interested in learning the technical details about computers? (e.g. computer hardware, engineering, networking etc.)	77.8	4.4	2.2	15.6	45	0.07	
Are you interested in learning the applications of computers? (e.g. Artificial Intelligence, Database design etc.)	82.2	0.0	2.2	15.6	45	0.07	
Are you interested in learning the fundamentals of computer science? (e.g. algorithms, data structures etc.)	76.1	4.4	6.5	13.0	46	0.51	
Are you studying Computer Science to learn programming?	52.3	18.2	11.4	18.2	44	0.58	
Was Computer Science your first subject choice?	70.5	15.9	4.6	9.1	44	0.69	
Do you plan to major in Computer Science?	80.0	11.1	4.4	4.4	45	1.00	
Are you interested in learning how to use a computer? (e.g. word processing, spreadsheets etc.)	69.6	17.4	6.5	6.5	46	1.00	
Do you have a computer at home?	73.9	19.6	4.4	2.2	46	1.00	
Do you think that there are good jobs available for people with Computer Science degrees?	95.6	0.0	2.2	2.2	45	1.00	
Are you confident of using new technology? (e.g. VCR, computers, remote controls)	100	0.0	0.0	0.0	46		
Are you interested in learning about scientific computing? (e.g. DNA sequencing, 3D graphics, modelling / simulation etc.)	95.6	0.0	0.0	4.4	45		

Table 2
Computer Science content and interest

	Unchanged		*Changed*				
Before	*Yes*	*No*	*No*	*Yes*			
After	*Yes*	*No*	*Yes*	*No*	n	p	
Female	28.6	21.4	50.0	0.0	14	0.02	*
Male	65.5	13.8	17.2	3.5	29	0.22	
Used a computer without programming	30.8	23.1	46.2	0.0	13	0.03	*
Used a computer including programming	69.2	7.7	19.2	3.9	26	0.22	
Encouraged to study a science degree	48.0	12.0	36.0	4.0	25	0.02	*
Not encouraged to study a science degree	58.8	23.5	17.7	0.0	17	0.25	

Table 3
Further analysis: Do you have a clear idea of what Computer Science involves?

	Unchanged		*Changed*				
Before	*Yes*	*No*	*No*	*Yes*			
After	*Yes*	*No*	*Yes*	*No*	n	p	
Female	71.4	7.1	0.0	21.4	14	0.25	
Male	67.7	6.5	0.0	25.8	31	<0.01	*
Used a computer without programming	76.9	0.0	0.0	23.1	13	0.25	
Used a computer including programming	64.3	10.7	0.0	25.0	28	0.02	*
Encouraged to study a science degree	80.0	0.0	0.0	20.0	25	0.06	
Not encouraged to study a science degree	52.6	15.8	0.0	31.6	19	0.03	*

Table 4
Further analysis: Do you think Computer Science and mathematics are closely related?

	Unchanged			*Changed*				
	A	*N*	*D*	*Pos*	*Neg*	*n*	*p*	
Working with computers is boring	0.0	6.8	70.5	20.5	2.3	44	0.01	*
Computer Science is not interesting because it involves working with machines instead of people	0.0	2.2	71.7	21.7	4.4	46	0.02	*
It is difficult to find interesting jobs in computer science	0.0	0.0	75.0	20.0	5.0	40	0.05	
I am interested in learning to use computers to solve practical problems; not in learning about the computer itself	17.5	15.0	12.5	35.0	20.0	40	0.14	
Computer Science involves mainly programming	12.8	18.0	28.2	20.5	20.5	39	0.60	
There are many jobs for people who have studied computer science	66.7	0.0	0.0	16.7	16.7	36	0.61	
Computer Science work involves mainly word processing	0.0	0.0	83.8	8.1	8.1	37	0.66	

Table 5
Perceptions of Computer Science

	Possible jobs		*Personal goal*	
	Before	*After*	*Before*	*After*
All				
Don't know/no response	21.7	19.6	37.0	30.4
Programmer	67.4	71.7	30.4	26.1
Development/design	19.6	21.7	8.7	13.0
Networking-related	19.6	28.3	4.3	4.3
Analyst/system analyst	17.4	19.6	6.5	6.5
Web-related	8.7	13.0	2.2	8.7
System administration	6.5	10.9	0.0	2.2
Technician	4.3	13.0	2.2	2.2
Degree requirement unlikely	19.6	23.9	4.3	13.0

Table 6
Responses about which jobs one would be qualified for with a Computer Science degree, and which job the respondents would like personally

	Possible jobs		*Personal goal*	
	Before	*After*	*Before*	*After*
Female				
Don't know/no response	33.3	13.3	53.3	26.7
Programmer	53.3	66.7	20.0	13.3
Development/design	20.0	20.0	0.0	13.3
Networking-related	13.3	20.0	0.0	6.7
Analyst/system analyst	26.7	26.7	6.7	6.7
Web-related	13.3	33.3	6.7	20.0
Technician	0.0	13.3	0.0	6.7
Consultant	0.0	13.3	0.0	6.7
Database-related	0.0	13.3	0.0	13.3
Degree requirement unlikely	13.3	33.3	6.7	20.0
Male				
Don't know/no response	16.1	22.6	29.0	32.3
Programmer	74.2	74.2	35.5	32.3
Development/design	19.4	22.6	12.9	12.9
Networking-related	22.6	32.3	6.5	3.2
Analyst/system analyst	12.9	16.1	6.5	6.5
System administration	9.7	12.9	0.0	3.2
Technician	6.5	12.9	3.2	0.0
Research	12.9	0.0	12.9	0.0
Degree requirement unlikely	22.6	19.4	3.2	9.7

Table 7
Responses by gender about which jobs one would be qualified for with a Computer Science degree, and which job the respondents would like personally

	Before	*After*
All		
Don't know/no response	13.0	23.9
Programming	50.0	50.0
Hardware	17.4	10.9
How computers work	13.0	8.7
How to use a computer	10.9	0.0
Algorithms	4.3	15.2
Female		
Don't know/no response	26.7	40.0
Programming	60.0	53.3
Hardware	13.3	6.7
How to use a computer	20.0	0.0
Algorithms	13.3	6.7
Networks	13.3	6.7
Software	13.3	6.7
Programming languages	13.3	0.0
Male		
Don't know/no response	6.5	16.7
Programming	45.2	48.4
Hardware	19.4	12.9
How computers work	16.1	9.7
Algorithms	0.0	19.4

Table 8
Responses about expected Computer Science course content

	Yes		No	n
Were the Computer Science I courses what you expected?	33.3		66.7	42
	A	*N*	*D*	*n*
The CS I courses were enjoyable	34.8	45.7	17.4	46
The CS I courses were interesting	52.2	37.0	8.7	46
The CS I courses were challenging	67.4	28.3	4.4	46
The CS I courses were suitable preparation for a career as a Computer Science professional	34.8	39.1	8.7	46
The CS I courses were useful for other courses at university	28.3	39.1	23.9	46
I learned a lot of facts from the CS I courses	60.9	28.3	6.5	46
My ability to do independent study improved during the CS I courses	41.3	45.7	8.7	46

Table 9
Opinions about the course