Does A "Pre-Introduction" Course Improve Subsequent Performance In "Introduction To Computer Programming"?

John Kevin Doyle, (E-mail: kdoyle@ben.edu), Benedictine University

ABSTRACT

A new Computer Science Introduction course is described. Student grades and the withdraw rate for the next course are both improved, and the proportion of student Computer Science majors is increased.

Initial results indicate that the availability of the new Introduction to Computer Science (CSO) reduces the number of students who withdraw from the old Introduction to Computer Science (CS1) almost certainly because students who simply need an introduction to computers no longer sign up for CS1 – a computer science and computer programming course). The limited data for students who took CS1 after completing CS0 do not as yet statistically support the hypothesis that CS0 will improve performance in CS1. A higher proportion of students who completed both CS0 and CS1 declared Computer Science or Information Systems majors, than the proportion of those who just completed CS1.

1. INTRODUCTION

For some years, our Computer Science and Information Systems department has offered a CS1 course as the introductory course for both Computer Science and Information Systems majors. This course was essentially a programming course, and introduced students to computer science and information systems from that perspective. From 2001 to 2003, the cumulative "success" rate in this course (i.e., the non-"DWF" rate) was 60%. We believed this low number to be due to inadequate preparation of the students, and to the fact that some other programs (e.g., Nuclear Medicine Technology) required this course of their majors as a way to introduce the students to computers. CS1 was neither targeted at nor appropriate for these students.

In fall 2003, to address both concerns, the department introduced a new course, "Introduction to Computer Science" (which we will refer to as CS0 in this paper) and made it a pre-requisite for CS1. Students who simply needed an introduction to computers could take CS0 as a stand-alone course. We expected that students who took CS0 would also be better prepared for CS1, and hence have a higher success rate. During the first transitional year, students in spring 2004 and summer 2004 were permitted to register for CS1 with or without having completed CS0. This paper describes the new course, and analyzes the initial year's experience.

2. PREVIOUS WORK

In [13], David Valentine investigated the number of presentations dealing with the First Year Computer Science course in the *SIGCSE Technical Symposiums Proceedings* over the last twenty years. This is the CS1/CS2 course in Curriculum 78 [1]. Valentine showed that the number of papers at the TSPs has increased at a statistically significant rate, and also that the proportion of such papers dealing with the First Year has also increased. Valentine classifies such papers into Marco Polo presentations ("I went there and I saw this"), presentations on Tools used in First Year courses, presentations on Nifty assignments, presentations on John Henry courses (such a difficult course

that only a super-teacher could succeed), presentations on First Year course Philosophy, and finally presentations reporting Experimental results. This paper falls in the last category.

Most First Year papers focus on the CS1 and CS2 courses themselves. This paper instead reports the introduction of a pre-CS1 course, which has been studied less frequently.

Among the papers which reported pre-CS1 courses, Campbell [4] reports on a pre-CS1 course, targeted at "at risk" students, and primarily focused on programming. Her pre-CS1 course also served as a filter, in that a significant proportion of the students did not subsequently attempt the required course. In [8], Linder and Stell discuss a pre-CS1 introductory survey course as the first course for Computer Science majors and examine its impact on CS1 performance, as well performance in upper level courses throughout the Computer Science major.

Moskal, Lurie and Cooper [9] discuss "the Alice course" as a pre-CS1 course and report that it appeared to allow at-risk students to perform comparably as non-at-risk students. Baldwin, Scragg and Kooman [2] report a two-course introduction to the science of computing, and report comparable retention rates.

3. INTRODUCTION TO COMPUTER PROGRAMMING (CS1)

Our current CS1 course is essentially that recommended by the ACM/IEEE Computing Curriculum reports. Specifically, CC2001 [6] describes CS111I. Introduction to Programming as follows. This is essentially the same as the course which was called CS1 in Curriculum 78 [1].

Introduces the fundamental techniques of programming as a foundation for more advanced study of computer science. Considerable attention is devoted to developing effective software engineering practice, emphasizing such principles as design, decomposition, encapsulation, procedural abstraction, testing, and software reuse. Topics include standard programming constructs, problem-solving strategies, the concept of an algorithm, and fundamental data structures (strings, arrays, and records) along with an introduction to machine representation, graphics, and networking.

No programming or computer science experience or prerequisite is required. Students should have sufficient facility with high-school mathematics to solve simple linear equations and to appreciate the use of mathematical notation and formalism.

4. INTRODUCTION TO COMPUTER SCIENCE (CS0)

About 50% of our CS0 course is descriptive in nature, focusing on the use of computers. This portion of the course is delivered via readings, case studies and discussion. The other half of the course is an introduction to computer programming, using Visual Basic. The descriptive and programming aspects of the course are interleaved and to some extent interrelated. Our initial two offerings of CS0 used [12] and [7] as texts, and the third offering used [12] and [3].

The course catalog description of this course is "Techniques and theory of computer science and information systems in a rapidly changing technical environment. Technology topics include hardware, software, communications, databases, emerging technologies, internet and intranet. Information systems topics include information processing concepts and functional systems used in business. Provides programming fundamentals, with applications developed in a high-level language. Programming topics include variables, formatted input/output, arrays, looping, conditional execution, subroutines and functions."

This course thus includes an introduction to the basics of programming, but much less than in CS1. In addition, CS0 is much more focused on the uses of computers and information in our society than is CS1.

The programming component of CS0 begins with basic programming concepts and writing a simple Visual Basic application. We then introduce variables, constants and built-in functions in calculations. Next, we discuss decisions in programs (if/then, etc.), and then looping and lists. We discuss elementary file access, dialog boxes, error handling, and menus. Then, we introduce user-defined sub-procedures and functions, and finally arrays, structures and collections. All of these topics are introduced at a basic level, and relatively briefly. The student programming assignments are relatively simple, and principally involving the student modifying a supplied sample program.

The descriptive component of CS0 includes discussion of security, privacy and ethical issues (this is done first, since the students have seen much of this in the commercial media, so have knowledge and opinions to encourage informed discussion). We then discuss information systems in organizations, organizing data and information, telecommunications, the Internet, intranets and extranets, electronic commerce and transaction processing systems, information and decision support systems, artificial intelligence and expert systems, and virtual reality. We conclude with a discussion of systems development.

The introduction of CS0 is also part of our more general move from the two course introductory sequence, CS1 and CS2, recommended in Curriculum 78 [1], to the three course sequence strongly endorsed in CC 2001 [6], CS101I, CS102I and CS103I.

5. RESULTS

Table 1 summarizes the student performance (i.e., the grades earned) in the semesters under study. Our comparison is between the semesters before CS0 was introduced (fall 2000 – fall 2003) and the semesters after it was introduced (spring 2004 – summer 2004). Within the latter group, we compare results for students who had previously taken CS0 with those who had not done so; the results are summarized in Table 2.

In addition, we examined the number of students who completed CS1, either successfully or unsuccessfully, who are currently Computer Science or Information Systems majors or (for those who have graduated) who were Computer Science or Information Systems majors at graduation. We were interested in seeing if students' successful completion of CS0 made for a larger proportion of CS1 students becoming Computer Science or Information Systems majors. For this analysis, we viewed earning A, B, or C in the course as the measure of success. Table 3 shows the results for semesters before CS0 was introduced, Table 4 the results after CS0, and Table 5 the comparison of the Table 4 results for students who successfully completed CS0 and those who did not.

Table 1: Comparison of CS1 results before and after CS0

	Fall 2000 – Fall 2003		**Spring 2004 Summer 2004**	
	raw	**%**	**raw**	**%**
A	35	22.6%	16	48.5%
B	31	20.0%	7	21.2%
C	27	17.4%	1	3.0%
D	13	8.4%		
F	10	6.5%	2	6.1%
W	39	25.2%	7	21.2%
Totals	155		33	

Table 2: Comparison of CS1 results from students who did and did not complete CS0

	Spring 2004 – Summer 2004; students who did not previously take CS0		Spring 2004 – Summer 2004; students who previously took CS0	
	raw	%	raw	%
A	7	22.6%	9	52.9%
B	3	20.0%	4	23.5%
C	1	17.4%		
D		8.4%		
F		6.5%	2	11.8%
W	5	25.2%	2	11.8%
Totals	16		17	

Table 3: Declared major for students who completed CS1, vs. grade in CS1; fall 2000 – fall 2003

	CS	IS	Other	Total	CS+IS %
A	16	4	15	35	57.1%
B	9	4	18	31	41.9%
C	6	7	14	27	48.1%
D	4	4	5	13	61.5%
F	4	1	5	10	50.0%
W	6	7	26	39	33.3%
Totals	45	27	83	155	46.5%

Table 4: Declared major for students who completed CS1 vs. grade in CS1; spring 2004 – summer 2004

	CS	IS	Other	Total	CS+IS %
A	8	3	5	16	68.8%
B	3	0	4	7	42.9%
C	1	0	0	1	100.0%
D					
F	1	1	0	2	100.0%
W	3		4	7	42.9%
Totals	16	4	13	33	60.6%

Table 5: Declared major for students who completed both CS0 and CS1, vs. grade in CS1; spring 2004 – summer 2004.

	CS	IS	Other	Total	CS+IS %
A	6	1	2	9	77.8%
B	3		1	4	75.0%
C					
D					
F	1	1		2	100.0%
W	2			2	100.0%
Totals	12	2	3	17	82.4%

6. ANALYSIS AND CONCLUSIONS

Our first measure of success is earning a grade of A, B, or C in the course (and thus the complement is earning D or F, or withdrawing – the DWF category). Our overall success rate in CS1 has significantly improved,

and this is somewhat attributable to CS0. The overall success rate in CS1 in spring and summer 2004 improved from 60.0% (the percentage of students who earned A, B or C in fall 2000 – fall 2003; see Table 1) to 72.7% (the percentage of students who DWF in spring 2004 – Summer 2004, also from Table 1). This difference is significant at the p=0.085 level.

From Table 2, we see that the success rate in spring and summer 2004 among the students who had previously completed CS0 was 76.4%, somewhat higher than the 73.3% success rate for students who did not previously take CS0. This difference is not statistically significant.

A direct comparison of student performance (the proportion of students who earned A, B or C) in Spring and Summer 2004, between those who had previously completed CS0 and those who had not, only weakly supports the hypothesis that student performance in CS1 is improved among those who previously completed CS0. This difference is significant at only the p=0.309 level.

An alternative measure of success in CS1 is the proportion of students who earned A or B in the course. Using this definition of success, we compare the 42.6% success rate for all CS1 students in fall 2000 – fall 2003 (Table 1) with the 76.4% success rate among 2004 – summer 2004 students who completed CS0 before CS1 (Table 2). This difference is significant at the p=0.004 level (so it is statistically significant), and thus provides good evidence that CS0 improves subsequent success in CS1.

The withdraw rate from CS1 is much improved among students who previously completed CS0. That is, the proportion of students who withdrew (25.2% in Table 1 in fall 2000 – fall 2003) is reduced to 11.8% in Table 2 in spring and summer 2004, among the students who had previously completed CS0. This difference is significant at the p=0.109 level.

Initial results therefore indicate that the availability of CS0 reduces the number of students who withdraw from CS1, probably because students who simply need an introduction to computers no longer sign up for CS1.

The proportion of students who declared Computer Science or Information Systems majors has also been improved by the introduction of CS0, as seen by comparison of Tables 3, 4 and 5. There is an overall increase in the proportion of CS1 students who became CS or IS majors (46.5% to 60.6%); see Tables 3 and 4. This difference is significant at the p=0.069 level.

The proportion of students who declared Computer Science or Information Systems majors is even higher for those students who completed CS0 (82.4%). This difference is significant at the p=0.002 level (and thus statistically significant).

Of course, the absolute number of students who declared Computer Science or Information Systems majors is affected by other trends in Computer Science education (the overall decline in Computer Science and Information Systems majors over the last five years). But, we are studying the proportion of CS1 students who declare Computer Science or Information Systems majors, not the number of such students.

Measuring this proportion immediately after a student completes CS1 (as was necessarily done in the spring 2004 – summer 2004 case) gives a somewhat inflated view of the proportion of majors. The likelihood of a student being a Computer Science or Information Systems major declines over time (as the courses become more complex, the number of students who continue declines). As Moskal, Lurie and Cooper [9] observe, "More than half the college students that initially declare a major in computer science change their major prior to graduation [5]", but that "the majority of students leave computer science by the end of freshman year [10]". Thus, the increase in proportion after CS1 was so significant, that we believe the result is real.

7. FUTURE WORK

We intend to continue to monitor the overall success/non-success rates in CS1, and hope/expect to see continued improvement. We will also continue to closely monitor the proportion of students completing CS1 who become and remain Computer Science and Information Systems majors.

We would also like to analyze prior experience of students entering CS1 (including high-school courses, prior college level courses, work, and hobby experience) to see whether there is correlation between success in CS0, prior experience and success in CS1. An initial attempt at this, with the survey shown in Figure 1 below, showed no strong correlations. Our current belief is that a better design of the survey may reveal some correlations. The wide variation of "prior experience" makes the survey design difficult.

8. ACKNOWLEDGEMENTS

Thanks to Dr. Daniel E. Nohl, the designer of our CS0 and the instructor for the first year. Thanks to Dr. David C. Sonnenberger, Benedictine's PeopleSoft guru, who helped extract the detailed student performance data.

9. REFERENCES

1. ACM Curriculum Committee on Computer Science, "Curriculum 78: Recommendations for the undergraduate program in computer science", *Communications of the ACM*, 22(3):147-166, March 1979.
2. Doug Baldwin, Greg Scragg, and Hans Koomen, "A three-fold introduction to computer science**",** *ACM SIGCSE Bulletin , Proceedings of the Twenty-fifth SIGCSE Symposium on Computer Science Education***,** Volume 26, Issue 1 (1994): 290-294.
3. Julia Case Bradley and Anita C Millspaugh, *Programming in Visual Basic .NET*, McGraw-Hill, 2003.
4. Patricia F. Campbell, "The effect of a preliminary programming and problem solving course on performance in a traditional programming course for computer science majors", Proceedings of the fifteenth SIGCSE Technical Symposium on Computer Science Education, 1984: 56-64.
5. Division of Research, Evaluation and Communication, Directorate for Education and Human Resources, *The Learning Curve: What We Are Discovering about U.S. Science and Mathematics Education*, L. E. Suter (ed), National Science Foundation, Washington, D.C. (1996); NSF 96-53.
6. The Joint Task Force on Computing Curricula, IEEE Computer Society, Association for Computing Machinery, "Computing Curricula 2001 Computer Science, Final Report", Dec. 15, 2001, http://www.computer.org/education/cc2001/cc2001.pdf.
7. Philip A. Koneman, *Visual Basic.NET Programming for Business*, Prentice-Hall (2004).
8. Jeffrey Linder and Stephen Sheel, "Session 1A: Computer science education: A critical view of a new computer science introductory course", Proceedings of the 30th Annual Southeast Regional Conference, Raleigh, NC, 1992: 385-388.
9. Barbara Moskal, Deborah Lurie, and Stephen Cooper, "Outcomes assessment: Evaluating the effectiveness of a new instructional approach*",* Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, Norfolk. March 3-7, 2004: 75-79.
10. Elaine Seymour and Nancy M. Hewitt, *Talking about Leaving, Why Undergraduates Leave the Sciences*, Westview Press, Boulder (1997).
11. Terry Sincich, *Statistics by Example*, 2nd. ed., Dellen Publishing Co., Dallas (1985): 430-434.
12. Ralph M. Stair and George W. Reynolds, *Fundamentals of Information Systems*, 2nd ed., Course Technology, 2003.
13. David W. Valentine, "CS Educational Research: A Meta-Analysis of SIGCSE Technical Symposium Proceedings", Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, Norfolk. March 3-7, 2004: 255-259.

Figure 1. "Prior Experience" Questionnaire.

Please check all of the following Computer Science courses which you completed before enrolling at Benedictine University (e.g., in high school, at a community college, at another university, …):

- ❑ Introduction to Computers
- ❑ Introduction to Microcomputer Applications
- ❑ Introduction to Word Processing
- ❑ Introduction to Spreadsheets
- ❑ Using the Web for Research
- ❑ Programming in Basic
- ❑ Programming in Pascal
- ❑ Programming in C++
- ❑ Programming in Java
- ❑ Introduction to Computer Science

Please note below any Computer Science courses which you completed before enrolling at Benedictine (e.g., in high school, at a community college, at another university, …), other than those listed above:

Please note below any non-academic computer experience you had, before you enrolled at Benedictine, which may have helped you in Computer Science courses at Benedictine:

Please check all of the following CIS/CMSC courses which you have completed at Benedictine University:

- ❑ CMSC 100 Introduction to PC Software
- ❑ CMSC 105 Microcomputer Applications to Teaching
- ❑ CIS/CMSC 120 Problem Solving with Computers
- ❑ CIS/CMSC 180 Introduction to Computer Science
- ❑ CIS/CMSC 200 Computer Programming
- ❑ CIS/CMSC 205 Data Structures and Algorithms

Thank you very much for your cooperation. If you are interested in receiving an email copy of our paper, please note your email address: _____

NOTES