

# What Effect Does CS0 Have On CS1 Performance And Major Retention?

John Kevin Doyle, (Email: [kdoyle@ben.edu](mailto:kdoyle@ben.edu)), Benedictine University

## ABSTRACT

*Three years ago, due to a lower than desired success rate in our first major course, Introduction to Computer Programming (CS1), and a lower than desired proportion of students in CS1 selecting Computer Science or Computer Information Systems as their major, we introduced a new course, Introduction to Computer Science (CS0), and made it a prerequisite for CS1. It was therefore required of all prospective Computer Science majors and Computer Information Systems majors. Our intention was to improve the subsequent performance of students in CS1, to reduce the proportion of students who withdrew from CS1, and to increase the proportion of CS1 students who became Computer Science or Computer Information Systems majors. This paper reports our progress on these measures.*

*Our results show that after CS0 was introduced, students performed better in CS1 than before CS0 was introduced, and that this difference is statistically significant. The withdraw rate from CS1, among students who completed CS0, has been reduced. The proportion of students who performed well (earned a grade of A or B) in CS1 has been raised among students who completed CS0, and this difference is statistically significant.*

*A higher proportion of students who completed both CS0 and CS1 declared Computer Science or Computer Information Systems majors, than before CS0 was introduced, and this difference is statistically significant. The proportion of students who performed well in CS1 and who declared Computer Science or Computer Information Systems majors was substantially raised, and this difference is statistically significant.*

*We compare these results with previous papers on this topic, and conclude that the results are consistent, and the statistical significance of the results is improved by the larger sample sizes being analyzed.*

## INTRODUCTION

Three years ago, in Fall 2003, we determined that the success rate in our first major course, Introduction to Computer Programming (CS1), was lower than desired. The proportion of students in CS1 selecting Computer Science or Computer Information Systems as their major was also lower than desired. We introduced a new course, Introduction to Computer Science (CS0), made it a prerequisite for CS1, and required it of all prospective Computer Science majors and Computer Information Systems majors. While we made CS0 a pre-requisite for CS1, we also publicized that students who simply needed an introduction to computers could take CS0 as a stand-alone course. Our intention was to improve the subsequent performance of students in CS1, to reduce the proportion of students who withdrew from CS1, and to increase the proportion of CS1 students who became Computer Science or Computer Information Systems majors. This paper reports our progress on these measures.

## **PREVIOUS WORK**

In [3], David Valentine examined papers dealing with the First Year Computer Science course in the *SIGCSE Technical Symposiums Proceedings* over the last twenty years, and classified 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 present paper falls in Valentine’s last category.

The author’s initial paper on this topic [4] reported on other published research on pre-CS1 courses. Both [4] and [5] reported initial results in this study. The current paper builds on these papers, by extending the period of study, and adding additional analysis.

### **Introduction to Computer Programming (CS1)**

Essentially the same CS1 course has been taught over the entire study period (Fall 2000 to Spring 2005). The instructors and textbooks changed, but the course description, objectives, and method of instruction remained essentially the same. The course description is “An introduction to software design, algorithm development and implementation with the C++ programming language. Elementary programming structures, functions, and text and file processing. Functional and object-oriented design and programming, real world and application modeling, testing and debugging. Prerequisites: [CS0 and a math course].”

This is essentially the CS1111 course recommended in CC2001 [2]: “Introduction to Programming: 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. *Prerequisites:* No programming or computer science experience 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.”

### **Introduction to Computer Science (CS0)**

Our CS0 course is half an introduction to computer programming, using Visual Basic, and half descriptive, focusing on the use of computers. The programming and descriptive aspects of the course are interleaved and interrelated. CS0 is significantly more focused on the uses of computers and information in our society than is CS1, and significantly less focused on programming than is CS1.

The catalog description of this course is “Techniques and theory of information systems in a rapidly changing technical environment. Technology topics include hardware, software, communications, databases, emerging technologies, internet, 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, functions. [No prerequisites].” A more detailed description of CS0 is provided in [4].

## **RESULTS**

Table 1 summarizes the student grades in CS1 in the periods 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 – Spring 2005). Within the latter group, we compare CS1 grades for students who had previously taken CS0 with those who had not done so; the results are summarized in Table 2.

We also examined the number of students who completed CS1, either successfully or unsuccessfully, whose declared major is Computer Science or Computer Information Systems. We were interested in seeing if students' successful completion of CS0 made for a larger proportion of CS1 students becoming Computer Science or Computer Information Systems majors. Table 3 shows the results for semesters before CS0 was introduced, Table 4 the results after CS0, and Table 5 the results for those students who completed CS0 before CS1.

**Table 1: Comparison Of CS1 Grades Before And After CS0 Was Introduced**

	Fall 2000 – Fall 2003		Spring 2004 – Spring 2005	
	raw	%	raw	%
A	30	23%	22	49%
B	25	20%	12	27%
C	24	19%	2	4%
D	10	8%		
F	10	8%	2	4%
W	29	23%	7	16%
Totals	128		45	

**Table 2: Comparison Of CS1 Grades For Students Who Did And Did Not Complete CS0**

	Spring 2004 – Spring 2005; Students Who Had Not Previously Taken CS0		Spring 2004 – Spring 2005; Students Who Had Previously Taken CS0	
	raw	%	raw	%
A	8	57%	14	45%
B	3	21%	9	29%
C	1	7%	1	3%
D				
F			2	6%
W	2	14%	5	16%
Totals	14		31	

**Table 3: Declared Major For CS1 Students, Vs. Grade In CS1; Fall 2000 – Fall 2003**

	CS	CIS	Other	Total	CS+CIS %
A	17	3	10	30	67%
B	7	3	15	25	40%
C	6	7	11	24	54%
D	3	5	2	10	80%
F	4	1	5	10	50%
W	2	7	20	29	31%
Totals	39	26	63	128	51%

**Table 4: Declared Major For CS1 Students, Vs. Grade In CS1; Spring 2004 – Spring 2005**

	CS	CIS	Other	Total	CS+CIS %
A	10	7	5	22	77%
B	5	2	5	12	58%
C	2			2	100%
D					
F	1	1		2	100%

W	2	1	4	7	43%
Totals	20	11	14	45	69%

**Table 5: Declared Major For CS1 Students Who First Completed CS0, Vs. Grade In CS1; Spring 2004 – Spring 2005**

	CS	CIS	Other	Total	CS+CIS %
A	8	4	2	14	86%
B	5	2	2	9	78%
C	1			1	100%
D					
F	1	1		2	100%
W	2	1	2	5	60%
Totals	17	8	6	31	81%

## ANALYSIS AND CONCLUSIONS

We examine several different metrics to determine whether the introduction of CS0 improved student performance in CS1. The first success metric is earning a grade of A, B, or C in CS1 (and thus the complement is earning D or F, or withdrawing – the DWF category). By this metric our overall success rate in CS1 has significantly improved, and this is somewhat attributable to CS0. The overall success rate in CS1 improved from 62% (the percentage of students who earned A, B or C in CS1 in Fall 2000 – Fall 2003; see Table 1) to 80% (the percentage of students who earned A, B or C in Spring 2004 – Spring 2005, also from Table 1). This difference is statistically significant ( $p = 0.004$ ). The 80% success rate is better the 73% success rate reported in [4] and the 78% success rate reported in [5].

Another success metric is earning a grade of A or B in the course (with the complement including DWF and C). Using this definition of success, we compare the 43% success rate for all CS1 students in Fall 2000 – Fall 2003 (Table 1) with the 74% success rate among Spring 2004 – Spring 2005 students who completed CS0 before CS1 (Table 2). This difference is significant at the  $p=0.000$  level (statistically significant). This result matches the corresponding results reported in [4] and [5], and on a longer study interval.

A third success metric is the withdraw rate (or actually the complement of the withdraw rate). The withdraw rate from CS1 is improved among students who previously completed CS0. That is, the proportion of students who withdrew from CS1 (23% in Fall 2000 – Fall 2003; from Table 1) is reduced to 16% in Spring 2004 – Spring 2005 CS1 students who had completed CS0 (Table 2). While not statistically significant ( $p=0.213$ ), this correlation is a positive sign. Again, this result matches the corresponding results reported in [4] and [5], and over a longer study interval.

The proportion of students who declared Computer Science or Computer 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 CIS majors (51% to 69%); see Tables 3 and 4. This difference is statistically significant ( $p=0.018$ ). This is a more positive result than the corresponding result reported in [4] – 61% and [5] – 64%. The proportion of all students who declared Computer Science or Computer Information Systems majors is even higher for those students who completed CS0 (81%). This difference is significant at the  $p=0.001$  level (and thus statistically significant).

Among high-performing CS1 students (those who earned A or B in CS1), the proportion of students who declared Computer Science or Computer Information Systems majors moved from 55% in Fall 2000 – Fall 2003 (Table 3) to 71% in Spring 2004 – Spring 2005 (Table 4), significant at the  $p=0.066$  level. However, examining only the students who completed CS0 and who later earned A or B in CS1, the proportion who declared Computer Science majors rose to 83%. This difference is statistically significant ( $p=0.007$ ).

The actual number of students who declared Computer Science or Computer Information Systems majors is affected by other trends in Computer Science education (the overall decline in Computer Science and Computer Information Systems majors over the last five years, possibly due to the .com bubble, the telecomm implosion, and off-shoring). But, we are studying the proportion of CS1 students who declare Computer Science or Computer Information Systems majors, not the absolute number of such students.

## **TECHNICAL BACKGROUND**

In 2000, Benedictine University moved academic records to an SQL server fronted by the PeopleSoft ERP system. All student records for students then enrolled were moved to the new system, and all subsequent student records were created in the new system. It is therefore possible to use PeopleSoft queries to extract records for all students who have graduated or terminated since the transition, and for all currently enrolled students.

We extracted records for all students in all Computer Science (CMSC) and Computer Information Systems (CIS) courses. The fields in the extracted records were Student Name, Term Code, Term Name, Course Department, Course Number, Course Section, Grade, and Major. For ease of manipulation, the records were delivered in an Excel spreadsheet.

To assist in the comparisons reported in this paper, a field was added to each record in the Excel spreadsheet, with value true for a single earliest record associated with a single student and false for all subsequent records associated with that student. This allowed, for example, counting the number of Computer Science majors by counting each student only once. Another field was added with value true for all courses taken by a student after they successfully completed CS0, and false for all other courses for that student (and false for all courses for students who did not successfully complete CS0).

Then, the data for the first comparison (earning a grade of A, B, or C in CS1 vs. DWF, comparing Fall 2000 – Fall 2003 vs. Spring 2004 – Spring 2005) was obtained by first filtering for grades in Course Department = CMSC or CIS and Course Number = 200, then filtering Term  $\leq$  Fall 2003, and finally counting the number of grades ABC vs. DWF. This was then repeated for Spring 2004  $\leq$  Term  $\leq$  Spring 2005. The resulting data is that reported in Table 1, and was used to generate the first comparison.

## **FUTURE DIRECTIONS**

We would like to use the techniques discussed in this paper for an initial assessment of student success in three different delivery modes of Benedictine University's MBA degree program. The delivery modes are first, a traditional on-site program at our main campus in Lisle, IL; second, a fully online *WebFlex* program; and third, a traditional on-site program at Shenyang University in Shenyang, China.

The variables we intend to examine include course delivery mode, student grade (both in core MBA courses, and in advanced MBA courses), and student gender. If the data is available, we would also like to examine the influence of student age, and of student professional experience.

A portion of this proposed study will focus on identifying additional student, course, and program characteristics which should be incorporated in future studies, how to obtain these characteristics in a cost effective manner, and how to record them in a manner which makes future investigation efficient.

## **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 student grade and major data. Thanks to Dr. Ralph D. Meeker and Dr. Nohl for several helpful conversations on this study.

**REFERENCES**

1. Terry Sincich, *Statistics by Example*, 2<sup>nd</sup>. ed., Dellen Publishing Co., Dallas (1985): 430-434.
2. 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>.
3. David W. Valentine, CS Educational Research: A Meta-Analysis of SIGCSE Technical Symposium Proceedings, Proceedings of the 35<sup>th</sup> SIGCSE Technical Symposium on Computer Science Education, Norfolk. March 3-7, 2004: 255-259.
4. John Kevin Doyle, Does a 'Pre-Introduction' Course Improve Subsequent Performance in 'Introduction to Computer Programming'?, *Journal of College Teaching & Learning*, (April 2005): 27-33.
5. John Kevin Doyle, Improving Performance and Retention in CS1, *Journal of Computing Science in Colleges*, Vol. 21, No. 1 (October 2005): 11-18.

**NOTES**