Contributing to Success in an Introductory Computer Science Course: A Study of Twelve Factors

Brenda Cantwell Wilson
Department of Computer Science
Murray State University
Murray, Ky 42071
brenda.wilson@murraystate.edu

Sharon Shrock
Department of Curriculum & Instruction
Southern Illinois University
Carbondale, IL 62901
sashrock@siu.edu

Abstract

This study was conducted to determine factors that promote success in an introductory college computer science course. The model included twelve possible predictive factors including math background, attribution for success/failure (luck, effort, difficulty of task, and ability), domain specific self-efficacy, encouragement, comfort level in the course, work style preference, previous programming experience, previous non-programming computer experience, and gender. Subjects included 105 students enrolled in a CS1 introductory computer science course at a midwestern university. The study revealed three predictive factors in the following order of importance: comfort level, math, and attribution to luck for success/failure. Comfort level and math background were found to have a positive influence on success, whereas attribution to luck had a negative influence. The study also revealed by considering the different types of previous computer experiences (including formal programming class, self-initiated programming, internet use, game playing, and productivity software use) that both a formal class in programming and game playing were predictive of success. Formal training had a positive influence and games a negative influence on class grade.

1 Introduction

Numerous studies of success in computer science including various previous computing experiences as possible predictors [2, 3, 7, 11, 12] have been conducted. Factors such as work style preference [4] and self-efficacy [1, 9, 10] as predictors have also been studied. Although attribution theory has been included in studies of other disciplines [3, 5, 6, 8], few studies have used the theory as a basis for explaining success in computer science. Attribution theory involves explanations that people give for their successes and failures. The explanations can be of a stable nature (attributing outcome to ability or difficulty of task) or an unstable nature (attributing outcome to luck or effort). The theory suggests that when people attribute their successes to unstable causes (luck or effort) and their failures to stable causes (ability or task difficulty), the probability of persistence is low.

This study was different; it included all of these factors plus other factors such as encouragement to study computer science and comfort level in the computer science course in an effort to explain success. Success was operationalized as midterm course grade. (Because of the high attrition rates in introductory computer science courses and because of the desire to study this phenomenon as it relates to the factors contributing to success in the introductory computer science course, midterm grades were used to determine success in the course to enable the inclusion of the students who drop out of the course before the end of the semester.)

2 Methodology

The study attempted to determine what relationship exists between the factors of previous programming experience, previous non-programming experience, attribution for success/failure, self-efficacy, comfort level, encouragement from others, work style preference, math background, and gender of introductory computer programming students and their midterm course grade. Also, the study sought to discover which of the above mentioned factors are predictive of midterm course grade. Also, a look at different types of previous computing experiences to determine whether they are predictive of success in CS1 was included.

2.1 Subjects

Approximately 130 students were enrolled in six sections of CS 202 Introduction to Computer Science at a comprehensive Midwestern university (approximately 22,000 student population) during the spring of 2000.

SIGCSE 2001 2/01 Charlotte, NC, USA
 ISBN 1-58113-329-4/01/0002...$5.00

There were 105 students who voluntarily participated in the study. CS 202 is the first programming class required in the computer science major and uses C++ as the programming language.

2.2 Instruments

Two instruments were used to collect data from the subjects: a questionnaire and the Computer Programming Self-Efficacy Scale developed by Ramalingam & Wiedenbeck [10]. The questionnaire collected data on the following items: gender, math background (number of semesters of high school math classes taken), previous programming experiences, previous non-programming computer experiences, encouragement by others to pursue computer science as a career, comfort level, work style preference, and attribution for perceived "success" or "failure" on the midterm exam. A pilot test was given to enable the researcher to find any ambiguities in the questionnaire, and revisions were made appropriately. One expert in the field of psychology research and two experts in the field of testing and evaluation were asked to evaluate the face validity of the questionnaire. Four seasoned computer science professors examined the content of the instrument. The questionnaire was found to have high content validity for measuring the variables in the study. A test-retest was used to examine the reliability of the questionnaire. The instrument was administered to students in two sections of an introductory computer science course at another university. Because the questionnaire was intended to measure different attributes, it was necessary to determine eight correlations. The Pearson Correlation coefficients were .98 for math background, 1.0 for previous programming course, .72 for previous self-initiated programming experience, .95 for previous non-programming experience, .80 for work style preference, .88 for comfort level, .72 for attributions to success/failure, and 1.0 for encouragement. The Computer Programming Self-Efficacy Scale was used to collect data on domain-specific self-efficacy as it relates to tasks in the C++ programming language. The authors reported an overall alpha reliability of .98 on the instrument.

2.3 Predictor Variables

Twelve predictor variables were included in the study. The way in which each variable was measured is described below: (all of this data except for self-efficacy, and midterm course grade were collected via the questionnaire)

1. Gender – a dichotomous variable (male or female).

2. Previous programming experience – a dichotomous variable determined by whether the subject had engaged in any programming prior to the course.

   In order to study the types of previous programming experience, this variable was subdivided into two areas: (a) formal programming course taken – a dichotomous variable determined by whether the subject had a previous programming course or not and (b) self-initiated programming experience – a dichotomous variable determined by whether the subject learned to write programs separate and apart from a formal class.

3. Previous non-programming experience – a dichotomous variable determined by whether the subject had engaged in non-programming computer activities.

   In order to study the types of previous non-programming experience, this variable was subdivided into three areas: (a) internet experience – a continuous variable determined by the number of hours per week each subject reported using the internet, e-mail, chat-rooms, and/or on-line discussion groups, (b) games – a continuous variable determined by the number of hours per week the subject reported spending time playing games on the computer, and (c) use of productivity software – a continuous variable determined by the number of hours per week the subject reported using productivity software such as word processing, spreadsheets, databases, and/or presentation software packages.

4. Encouragement to pursue computer science – a dichotomous variable representing whether the subject received encouragement to pursue computer science or not.

5. Comfort level – a continuous variable derived from seven questions on the questionnaire regarding asking and answering questions in class, in lab, and during office hours; anxiety level while working on computer assignments; perceived difficulty of course; perceived understanding of concepts in the course as compared with classmates; and perceived difficulty of completing the programming assignments. Numbers were appropriately assigned to the choices of the BAR (Behaviorally Anchored Rating) scale in these questions and summed for a composite score of comfort level.

6. Work style preference– a dichotomous variable (competitive or cooperative) representing the answer to a question about preference for writing computer programs and/or studying for exams.

7. Attributions – four continuous variables derived from the subject's rating of possible reasons for success or failure on the midterm exam. They are: (a) attribution to ability, (b) attribution to task ease/difficulty, (c) attribution to luck, and (d) attribution to effort.

8. Self-efficacy – a continuous variable, which is the summation of the choices made on a Likert-type scale from the Computer Programming Self-Efficacy Scale.

9. Math background – a continuous variable represented by the number of semesters of high school math courses reported by the subject.

2.4 Criterion Variable

The criterion variable of the study was the midterm grade in the introductory computer science class for each student. This was a continuous variable representing a number between 0 and 100.

To ascertain that the use of the midterm grade was a viable choice for determining success in the computer programming class, a correlation coefficient was generated using the midterm scores and the final scores in two sections of the first course in Computer Science from the fall semester of 1999. The Pearson Correlation Coefficient was extremely high and significant, $r = .97173$, $N = 48$, $p = .0001$, therefore, it seemed reasonable that the midterm grade was a good indicator of success in the class.

2.5 Procedure

During the spring semester the questionnaire and Computer Programming Self-Efficacy Scale were distributed after the exam and before midterm of the semester at a class lecture session. Data was collected from 105 students.

2.6 Analysis of Data

Although no study could be found that combined all of the predictor variables that are included in this study, some of the previous research could be used to determine an expected hierarchy of predictor variables. Therefore, based on the literature review and on the researcher's experience of teaching computer science, a hierarchical model was generated and tested using the general linear model. The model included twelve predictor variables in the following order: math, previous programming experience, attribution to luck, attribution to difficulty of task, comfort level, non-programming experience, work style preference, domain-specific self-efficacy, encouragement to study computer science, attribution to effort, attribution to ability, and gender. This model was tested and compared to the findings of the previous research studies in computer science success. All analyses used an alpha level of .05 to determine significance.

A residual plot was generated from the data confirming the multi-linear model. A correlation matrix was generated to examine how each of the 12 factors correlated with midterm grade and with each of the other predictor variables. By examining the R^2 and its p-value of the full-model regression equation, the proportion of variance in midterm grade accounted for by the twelve predictor variables was determined. The Type I sums of squares and Type III sums of squares with associated p-values were examined to determine the contribution of each factor over and above the other factors. The parameter estimates from the multiple regression tests were also examined to see whether each factor had a positive or negative effect on midterm grade. The full model for the twelve predictor variables was:

$Y = a_0U + a_1X1 + a_2X2 + a_3X3 + a_4X4 + a_5X5 + a_6X6 + a_7X7 + a_8X8 + a_9X9 + a_{10}X10 + a_{11}X11 + a_{12}X12 + E$

where Y = midterm course grade
X1 = 1 if previous programming; 0 otherwise
X2 = 1 if previous non-programming; 0 otherwise
X3 = rating for attribution to task difficulty
X4 = rating for attribution to luck
X5 = rating for attribution to effort
X6 = rating for attribution to ability
X7 = rating for self-efficacy
X8 = rating for comfort level
X9 = 1 if had encouragement; 0 otherwise
X10 = 1 if male; 0 if female
X11 = number of sem. of math courses
X12 = 0 if work style preference is cooperative; 1 if competitive
E = the errors of prediction

To determine if any of the previous computing experiences were predictive of success, a full model and four restricted models were used. The restricted models were constructed by dropping out one predictor variable from the full model. Each restricted model was tested against the full model to ascertain whether the contribution of each predicting factor over and above the other factors in combination was significant. The full model for previous computing experiences was:

$Y = b_0U + b_1P1 + b_2P2 + b_3P3 + b_4P4 + b_5P5 + + E$

where Y = midterm course grade
P1 = 1 if programming class; 0 otherwise
P2 = 1 if self-initiated programming; 0 otherwise
P3 = number of hours/week of Internet use
P4 = number of hours/week of games
P5 = number of hours/week of productivity software use
E = errors of prediction

3 Results

The proportion of variance in midterm score accounted for by the linear combination of the 12 factors was approximately .44, $R^2 = .4443$, which was statistically significant, $F(12, 92) = 6.13$, $p = .0001$. Three of the predictor variables contributed a significant difference in the midterm grade at the .05 level even after being considered last in the model. They were comfort level, math background, and attribution of success/failure to luck with p-values of .0002, .0050, and .0233 respectively. Two of the three significant predictive factors (comfort level and math) had positive correlations with the midterm score, but attribution of success/failure to luck had negative parameter estimation. (See Table 1.)

When stepwise multiple regression was used, two more variables showed significant influence in a five factor model. They were work style preference and attribution of success/failure to task difficulty. These five variables contributed to 40% of the variance. The work style preference was positively correlated to the midterm score, which indicated that an individual/competitive work style preference had a positive influence on the midterm score. Attribution to task difficulty was negatively correlated to midterm score.

Two of the previous computing experience variables showed significant influence in predicting the midterm score: previous programming course and games with p-values of .0006 and .0287 respectively. (See Table 2.) It was also noted that while the previous programming course variable had a positive influence on midterm grade, games

had a negative influence. Also the proportion of variance accounted for by the five previous programming and non-programming variables was .15 which was significant for the sample, p = .0041.

Table 1: Summary of Type I and Type III Sums of Squares General Linear Models Procedure
(R^2 = .444347, F = 6.13, P = .0001)

Source	DF	Type I SS	F	P
Math	1	2348.67	15.17	.0002
Prg	1	1203.42	7.77	.0064
Luck	1	2014.73	13.02	.0005
Diff	1	766.75	4.95	.0285
Cmf-lev	1	3460.21	22.36	.0001
NPrg	1	51.19	0.33	.5666
WPrf	1	625.34	4.04	.0474
SE	1	53.60	0.35	.5577
Enc	1	302.55	1.95	.1654
Effort	1	359.15	2.32	.1311
Ability	1	2.20	0.01	.9053
Gender	1	199.63	1.29	.2590

Source	DF	Type III SS	F	P
Math	1	1279.79	8.27	.0050
Prg	1	389.52	2.52	.1161
Luck	1	824.10	5.32	.0233
Diff	1	455.95	2.95	.0895
Cmf-lev	1	2334.38	15.08	.0002
NPrg	1	169.47	1.09	.2981
WkPrf	1	482.72	3.12	.0807
SE	1	12.53	0.08	.7767
Enc	1	170.09	1.10	.2973
Effort	1	296.74	1.92	.1695
Ability	1	3.42	0.02	.8821
Gender	1	199.63	1.29	.2590

4 Conclusion

Comfort level in the computer science class was the best predictor of success in the course. Math background was second in importance in predicting success in this computer science class. It is most interesting, in this study, that comfort level was found to be more important than math background. Most of the research studied for the literature review, which included math as a predictor, concluded that math and computer programming experience were the most important factors in success in computer science, although many of these studies did not include studying comfort level as such. Although programming experience (which included both a previous programming course and self-initiated programming) was not found to be significant in the full model, when the different types of computing experiences were compared as predictors of midterm grade, the previous programming course and game playing were both significant. It should be noted that the notion that game playing gives students an "edge" in a computer science course was not supported in this study. Game playing had a negative effect on the midterm grade.

The result for analysis of attribution to luck was also an interesting finding. To support most of the attribution research findings, attribution to luck would only be positively correlated to success in the course for those students who were unhappy with their score. In other words, if they could attribute their "low" score to an unstable cause such as luck, then they would continue to try to do better. In this study, however, attribution to luck for all students (whether happy or unhappy with their scores) was negatively correlated to midterm.

Table 2: Summary of Multiple Regression Analysis on Previous Programming and Non-Programming Experience

Analysis of Variance for Model				
Source	Df	R^2	F value	P
Model	5	.1577	3.706	.0041
Error	99			
C Total	104			
Parameter Estimates				
Variable	Parameter estimate	Standard error	T	P
Intercept	64.8755	2.6049	24.905	.0001
PrvPrgCs	10.6138	2.9944	3.545	.0006
SiPrg	1.8392	3.3532	.548	.5846
Int	.0906	.1765	.514	.6087
Games	-.4217	.1900	-2.219	.0287
Apps	.2315	.1761	1.315	.1917

Note. Variables
PrvPrgCs = previous programming course; SiPrg = self-initiated programming; Int = use of the Internet; Games = playing games on the computer; Apps = use of productivity software

5 Recommendations

Although this study did not show that higher comfort levels "cause" students to perform better in the computer science class, because of the positive correlation in this study between comfort level and success in the introductory computer science course, the notion that providing the optimum class environment for producing higher levels of comfort for students is at least warranted. It is suggested that professors of college computer science should understand the importance of providing an environment in the course which encourages students to ask and answer questions, both in class and outside of class, in a way that allows the students to feel comfortable and not intimidated. Opportunities for students to be able to consult with faculty, teaching assistants, or tutors were also indicated. The recent move in many universities to force students into large lecture sections for computer science, which by its very nature discourages dialogue between students and faculty, is an indication of the misunderstanding of the importance of the level of comfort students may need in this difficult discipline. Also, advisers should stress an appropriate mathematical background for students wanting

to pursue computer science. Finally, since attribution to luck showed a negative correlation with success, professors should endeavor to match class assignments and exam questions in the hope that students will not perceive luck as a reason for success or failure on the exams. Again, this suggestion is warranted even though the study only showed a negative correlation and not causation.

References

[1] Bandura, A. Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review, 84(2)*, (1977), 191-215.

[2] Bunderson, E.D., & Christensen, M.E.. An analysis of retention problems for female students in university computer science programs. *Journal of Research on Computing in Education 28(1)*, (1995), 1-15.

[3] Clarke, V.A., & Chambers, S.M. Gender-based factors in computing enrollments and achievement: Evidence from a study of tertiary students. *Journal of Educational Computing Research, 5(4)*, (1989), 409-429.

[4] Cottrell, J. I'm a stranger here myself: A consideration of women in computing. *Proceedings of the 1992 ACM SIGUCCS User Services Conference*, (1992), 71-76.

[5] Deboer, G.E. Factors related to the decision of men and women to continue taking science courses in college. *Journal of Research in Science Teaching, 21(3)*, (1984), 325-329.

[6] Dweck, C.S., & Leggett, E.L. A social-cognitive approach to motivation and personality. *Psychological Review (95)*, (1988), 256-273.

[7] Kersteen, Z.A., Linn, M.C., Clancey, M., & Hardyck, C. Previous experience and the learning of computer programming: The computer helps those who help themselves. *Journal of Educational Computing Research 4(3)*, (1988).

[8] Maeher, M.L. On doing well in science: Why Johnny no longer excels; why Sarah never did. In S.G. Paris, G.M. Olsen, & H.W. Stevenson (Eds.), *Learning and Motivation in the Classroom* (Chapter 8), (1988).

[9] Miura, I.T. The relationship of computer self-efficacy expectations to computer interest and course enrollment in college. *Sex Roles, 16(5)*, (1987), 303-311.

[10] Ramalingam, V., & Wiedenbeck, S. Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research, 19(4)*, (1998), 367-381.

[11] Taylor, H., & Mounfield, L. An analysis of success factors in college computer science: High school methodology is a key element. *Journal of Research on Computing in Education, 24(2)*, (1991), 240-245.

[12] Taylor, H., & Mounfield, L. Exploration of the relationship between prior computing experience and gender on success in college computer science. *Journal of Educational Computing Research 11(4)*, (1994), 291-306.