# Predictors of Success and Failure in a CS1 Course

## Nathan Rountree, Janet Rountree, and Anthony Robins

Department of Computer Science University of Otago {rountree,janet,anthony}@cs.otago.ac.nz

#### Abstract

We present the results of a survey which focuses on the backgrounds and expectations of a group of CS1 students in the first weeks of semester. When comparing their survey answers to their final grades on the course, we saw some surprising things: the group which indicated an intention to continue in computer science did no better than any other, and the strongest single indicator of success seems to be "expecting to get an A from the course."

### 1. Introduction

COMP103 is the first year programming course offered by the Department of Computer Science at the University of Otago. It is offered as an optional paper for non-computing majors, and is a prerequisite for continuing in the departments of Computer and Information Science. The course runs over thirteen weeks, and attracts around four hundred students.

In 2000 we changed our introductory programming language from Pascal to Java. Aside from the discussion this created about choice of language and what should be taught in the introductory curriculum, we started to wonder if there were any factors that could predict the pass/fail success of these students. For instance, did any previous course of study, or their level of enthusiasm for the taking the course instil a skill or attitude that could be seen as a factor in their success?

There is a wide variety of studies concerned with the improvement of computer science education. For CS1, there are still questions about *what* to teach—what paradigm [3], which language [4], and which features of particular languages [7] are essential components of the curriculum? Studies have also looked at the difficulties and consequences of the popular change in first language choice from procedural to object-oriented [2]. Initial interests for CS1 courses that use Java have tended to focus on the practicalities of teaching and resources available for establishing the course [1].

To our knowledge there are no published studies which investigate possible predictors of success and failure for novice students undertaking an introductory programming course. The study most closely related to our work was undertaken at IBM in New York [5], and looks at what factors may contribute to successful object-oriented learning. However, their subjects were professional programmers—new to object-orientation, but not to programming itself.

#### 2. The Study

Data was collected during the second semester of 2000 and again in 2001. In the first week of laboratory sessions, Students were asked to complete an optional online survey which mainly consisted of multi-choice questions. Replies were collected via email. For detailed information about the content of COMP103, see [6].

Information was requested regarding:

- status: gender, age, enrolment status (part or full-time), year of study at university, intended major, how keen they were to take COMP103;
- background: what recent mathematics courses they had taken, whether they felt their strongest background was in humanities, science, or commerce subjects, whether they knew any programming language(s) already;
- expectations: how difficult they anticipated the course would be, what they expected of the workload, what grade they expected to achieve, whether they intended to enrol in second-year computer science courses.

Any replies which were duplicates or did not have a valid student ID number were excluded. In addition, replies have only been included from students who submitted their second assessed exercise and/or who attended the final examination for the course. This decision was made in order to exclude any early withdrawals from COMP103, but to include those students who made an early commitment to the course and withdrew later. The total number of valid replies for 2000/2001 totalled 472, out of a possible 748.

Making the survey voluntary resulted in a self-selected sample that consisted of more than half the class, but also a clearly more successful group. However, we felt it was more important to have students answering the questions honestly rather than feeling coerced into providing information that they may not wish to share. We are making an assumption that the majority of students who answered the survey did so honestly, since there seemed to be genuine interest in providing the department with helpful information.

At the end of the semester, each student's reply was matched with their final mark out of 100. This mark reflects 30% worth of bi-weekly programming assignments, a single 20% mid-semester examination, and a 50% final examination. Students who answered the survey and completed the second laboratory exercise but were absent from the final exam have been counted as fails.

In COMP103 a student who receives a mark of 50% or greater achieves a pass. However, we generally find that students who achieve a mark of 70% or greater are likely to be more successful in subsequent computing papers, so we also report results in this range. From the 472 valid survey answers, 73% of the students received a passing grade, while 45% of the total managed to score over 70%.

Table 1 compares the passing and over 70 proportions of students who chose to answer the survey with those who did not:

Table 1:  $\chi^2$ -test result for "answered survey" vs. "did not answer"

	Answered	Did not Answer	$\chi^2$	df	р
Pass	73%	55%	26.02	1	< 0.001
Over 70	45%	29%	19.45	1	< 0.001

We conclude that the group of students willing to participate in the survey is more likely to do well in COMP103.

Since all questions on the survey were multi-choice, each of them was subjected to chi-square analysis to determine if deviations from the expected pass rate or 70+ rate were statistically significant. With an increase in  $\chi^2$ -value, the probability of the observed classes being independent of the explanatory variables diminishes. Table 2 summarises a comparison of the  $\chi^2$  values for each question in the survey, with respect to their pass rate and over 70 rate. Starred p-values are significant at 95% confidence and double starred values at 99% confidence.

Question	Categories	$\chi^2_{pass}$	$p_{\text{pass}}$	$\chi^2_{70+}$	p <sub>70+</sub>
gender	male, female	0.04	0.85	0.03	0.87
age	16–18, 19–21, 22-24, 25+	14.74	0.002**	6.6	0.086
full-time status	part-time, full-time	0.01	0.93	0.56	0.46
year	1 <sup>st</sup> , 2 <sup>nd</sup> , 3 <sup>rd</sup> , 4 <sup>th</sup> +	14.22	0.003**	8.36	0.039*
major	comp sci, info sci, comp & info sci, other	7.06	0.070	14.50	0.002**
keenness	extremely, fairly, neutral, not	13.26	0.004**	19.15	0.001**
recent math	school, uni, other, none	15.21	0.002**	11.54	0.009**
background	humanities, science, commerce	11.68	0.003**	35.77	0.001**
know other language(s)	no, yes	8.04	0.005**	13.65	0.001**
expected difficulty	easier (than my other papers), the same,	9.51	0.023*	25.14	0.001**
	harder, unsure				
expected workload	less (than my other papers), the same,	7.87	0.049*	7.69	0.053*
	more, unsure				
expected success	A grade, B grade, C grade, unsure/prefer	24.27	0.001**	44.51	0.001**
	not to say				
continuing in comp sci	yes, no, unsure	2.53	0.29	5.23	0.073

Table 2: Summary of  $\chi^2$ -test results on each survey question

There are some interesting results here:

- Pass rate and 70+ rate seem independent of gender, of whether a student is full-time or part-time, and of whether a student is intending to continue in computer science. The last was quite surprising, since we expected that the intention to continue would correlate strongly with success.
- The 70+ rate seems to be independent of age range, but the pass rate is not.
- The pass rate seems to be independent of intended major, but the 70+ rate is not.

## 3. Discussion

			1
	Answer	%≥50	%≥70
Age: Two things are surprising here: the unusually low pass rate of the 22-24	16–18	80	-
group, and the unusually high pass rate for 16–18. This overturned an expectation	19–21	72	-
that more mature students tend to do better at computer science papers. It is	22-24	54	-
possible that the 22-24 group is just old enough with respect to the New Zealand	25+	74	-
school system not to have had sufficient prior computing experience; however we			
consider this unlikely since a basic computing paper (COMP101) is a prerequisite			
to COMP103.			
Year of study: Clearly students in their second or third year of study at university	first year	79	48
do less well in COMP103 than students in their first or fourth year. Non-computing	second year	62	33
majors in their second or third year are presumably taking COMP103 to enhance	third year	61	39
their computing skills or to "fill-in" their points in order to complete a degree; there	fourth year+	72	53
were 37 students in this situation and 47% of them failed. Although we would need			
to look at individual student records to confirm this trend, we feel that this indicates			
that treating an introductory programming paper as a "filler" may not be a good			
idea.			
Intended major: Note that the pass rate has too high a p-value to be statistically	comp sci	-	55
significant. There is however a 20 percentage point difference between the	info sci	-	35

proportion of information science students achieving over 70% and the proportion	comp & info	-	39
of computer science students. We should point out that although COMP103 is a	other	-	42
prerequisite for continuing in both departments, there is very little material in the			
course that is specific to information science.			
Keenness: As we expected, that group of people who considered themselves to be	extremely	85	62
extremely keen to participate in COMP103 had a much higher pass rate than those	fairly	69	42
who were not. Less expected was the difference in pass rate between those who	neutral	78	38
were "fairly keen" (69%) as opposed to those who were "neutral" (78%). It would	not keen	60	27
seem as though being willing to "go all out" leads to success, but being just slightly			
reserved in your enthusiasm (enough to tick "fairly keen" rather than "extremely			
keen") is worse than reporting no feeling at all.			
Recent mathematics: Although COMP103 has no specifically mathematical	school	68	36
content found above year 10 in high-school, it does seem as though ability in	university	78	50
programming is suggested by ability in math. However, the disparity between	other	81	52
school math and university math is interesting. We suspect that it is the work habits	none	58	32
instilled by university math courses (small weekly assignments and laboratory			
sessions) that provide the advantage rather than simply the mathematical skills			
learned in the course.			
Background: Success rates are lower for students who consider themselves to have	humanities	61	28
a humanities background. This was of some concern, considering our effort to run a	science	79	57
course that did not rely on students' level of mathematical skill. We suspect that	commerce	70	31
there is a large difference in the style of COMP103 from what humanities students			
are used to. The rewards and frustrations inherent in programming are strongly			
polarised; thus there is a very different sense of what you have achieved and when			
you have achieved it. This may lead some students to an incorrect assessment of			
what they need to do to be successful in the course.			
Knowing a programming language already: Unsurprisingly, the 18% of students	no	71	41
who claimed to already know a programming language had a much higher success	yes	86	63
rate. However it is interesting to note that 12 of those 84 students still failed the			
course: we conclude that knowing a programming language is no guarantee of			
success in an introductory programming course.	11.007 1		
Expected difficulty, Expected workload: The last three questions all indicate that	more difficult	68	35
students are remarkably good at making their own assessment of the challenges	same	81	54
facing them and how well they are likely to react to them. In general, students who	easier	80	69
thought the course would be more difficult for them did less well than those who	undecided	69	42
thought it would be about the same as their other courses.	more work	64	39
	same	79	52
	less work	71	46
	undecided	58	42
Expected grade: This is clearly the strongest single indicator of success: students	A grade	90	70
who expect to get an A and are willing to say so are far more likely to be successful.	B grade	69	37
This suggests to us that a positive attitude is more important than having the right	C grade	63	24
background, and that students are fairly good at estimating their own ability.	unsure	64	38

# 4. Conclusion

We have presented the results of a survey taken over two semesters of an introductory programming course. The purpose was to see if there were factors independent of students' previous academic performance that influenced their success in the paper. We found that the strongest single indicator of success was the grade the student *expected* to achieve at the beginning of the course. The questions on expected grade, anticipated

difficulty, and anticipated workload indicate that students have a strong sense of how well they are likely to do within the first two weeks of the semester. Other factors that are related to success include whether students think their background is science, commerce, or humanities; whether they have recent university math experience; and what year of study they are in—but not always displaying the relationships we might expect.

Our results suggest several ways in which we might improve COMP103 before we even begin to consider curriculum or teaching methodology. Students should be told at the beginning of the course that they are remarkably good at assessing their own ability—that what they suppose they will get in this course will correlate strongly with what they will actually get.

Students should be given a clear message that taking introductory programming as a "filler" course is potentially a tactical error. We believe that the only good reason to take this course is because you want to learn how to program, and that the chances of failing the course increase dramatically when this is not true.

We need to be very precise at the beginning of the course about why COMP103 is likely to be different to other courses the participants may have done. It seems that the time commitment to laboratory sessions and programming assignments, the grappling with programming concepts, and the sharp distinction between an assignment that works and one that doesn't can come as an unpleasant surprise to some students.

Author Terry Pratchett has commented that really successful writers "...have to *want* to write. Too many people want to *have written*." We believe that there is a similar problem in learning to program: many people would like to have the skill, but find the mental attitude required to gain it is hard to sustain. Our results suggest that a positive attitude is the most important factor.

#### Acknowledgments

This work has been supported by internal University of Otago Research into Teaching grants. We are also grateful for the support of the COMP103 teaching team, especially Sandy Garner and Mike Atkinson.

#### References

- Andreae, P., Biddle, R., Dobbie, G., Gale, A., Miller, L., and Tempero, E. Experience Teaching CS1 with Java, *Journal of Computer Science Education*, 14, 1&2, 2000, 19–28.
- [2] Biddle, R. and Tempero, E. Java Pitfalls for Beginners, *SIGCSE Bulletin*, 30 (2), 1998, 48–52.
- [3] Deek, F., Kimmel, H., and McHugh, J. Pedagogical Changes in the Delivery of the First-Course in Computer Science: Problem Solving, then Programming, *Journal of Engineering Education*, 87 (3), 1998, 313–320.
- [4] Kolling, M. The Problem of Teaching Object-Oriented Programming, Part 1: Language, *Journal of Object-Oriented Programming*, 11 (8), 1999, 8–15.
- [5] Liu, C., Goetze, S., and Glynn, B. What Contributes to Successful Object-Oriented Learning, *OOPSLA* '92, 1992, 77–86.
- [6] Robins, A., Rountree, J., and Rountree, N. My Program is Correct but it Doesn't Run: a Review of Novice Programming and a Study of an Introductory Programming Paper. Technical Report OUCS-2001-06, 2001, University of Otago.
- [7] Stephenson, C., and West, T. Language Choice and Key Concepts in Introductory Computer Science Courses, *Journal of Research on Computing in Education*, 31 (1), 1998, 80–95.