Failure Rates in Introductory Programming

Jens Bennedsen

IT University West

Fuglesangs Allé 20

DK-8210 Aarhus V

Denmark

jbb@it-vest.dk

Michael E. Caspersen

Department of Computer Science

University of Aarhus

Aabogade 34, DK-8200 Aarhus N

Denmark

mec@daimi.au.dk

Abstract

It is a common conception that CS1 is a very difficult course and that failure rates are high. However, until now there has only been anecdotal evidence for this claim. This article reports on a survey among institutions around the world on failure rates in introductory programming courses. The article describes the design of the survey and the results. The number of institutions answering the call for data was unfortunately rather low, so it is difficult to make firm conclusions. It is our hope that this article can be the starting point for a systematic collection of data in order to find solid proof of the actual failure and pass rates of CS1.

Keywords: CS1, failure rate, pass rate, introductory programming.

1. Introduction

It is generally accepted that it is very difficult to learn to programming [7, 11, 14]. For example Bergin and Reilly [2] note that *It is well known in the Computer Science Education (CSE) community that students have difficulty with programming courses and this can result in high drop-out and failure rates.* (p. 293).

However, the belief that there generally are high drop-out and failure rates does not seem to originate from any official statistic or sound investigation of the subject —it is more in the realm of folk-wisdom, and claims that have been said so often that they are accepted as truths. The only source of information that we know of is authors who give pass-rates for their particular introductory course in articles describing other issues (e.g., [6]). We find it problematic not to have more solid evidence for this claim. Consequently, we have designed a study aiming at finding the average failure and pass rate for CS1 courses around the world.

False views on failure and pass rates can have serious implications for the quality of introductory programming courses. A lecturer with a high failure rate might accept that "this is just the way programming courses are since all programming courses have high failure rates" and consequently not take action to improve the course in order to reduce the failure rate.

Programming is one of the courses that students encounter first in their computer science university program. If these courses have as high failure rates as claimed, or if that rumour gets around to students, it could be one of the factors influencing the declining number of students taking a degree in computer science [13]. We therefore believe it is important to have more accurate numbers in order to pro-vide potential students with a better view of the difficulties of programming.

International organizations such as UNESCO collect data on worldwide educational activities. Their data is, however, not fine-grained enough to give numbers for individual courses.

This article describes the design and results of our study of failure rates in introductory programming courses (CS1 courses). The study has one major problem: only 63 institutions (12.7%) answered the call for information. This naturally influences the generalisability of our study; we hope that following studies will be able to provide more general conclusions.

2. Research

This section describes the research methodology: the research question, the questionnaire used to collect data, and the participants.

2.1 The Research Question

As mentioned in the introduction, it seems like high failure rates in introductory courses are the norm. However, no worldwide statistics on failure rates, drop-out rates, or pass rates for introductory programming courses at university level exist to back up this postulate. Our research question is therefore: *What are the failure and pass rates for introductory programming courses at university level?* And: *Is the failure rate high?*

2.2 The Questionnaire

In order to answer the research question, we developed a short, web-based questionnaire [4]. In the questionnaire, four terms were defined and the respondents asked to give numbers for

abort: the number of students aborting the course before the final exam

skip: the number of students not showing up for the final exam, but was allowed to

fail: the number of students who failed the course

pass: the number of students who passed the course

Apart from these numbers, we asked for the type of introductory course (imperative, object-oriented or functional [5]), the type of institution (university, college, etc.) and how the course was evaluated.

2.3 The Participants

The target group for this research is not an easily accessible group, so a selection of respondents is required. The group must contain universities and colleges teaching computer science from all over the world. To enable representatives from all over the world, the following five sources were chosen (respondents were addressed via email):

- The authors of articles for Koli Calling 2004, the 4th Annual Finnish / Baltic Sea Conference on Computer Science Education [12];

- The authors of articles for and the participants in panels at the 36th Technical Symposium on Computer Science Education [15];

- The authors of articles for and panel members at the 10th Annual Conference on Innovation and Technology in Computer Science Education [10];

- The authors of research papers at the 4th International Conference on Advanced Learning Technologies [8];

- The authors of articles at the Australian Computers in Education Conference [1].

We did not use the general SIGCSE mailing list since we have no knowledge of the geographical distribution of the recipients.

It is debatable whether the selected universities are representative. Another problem is whether the persons actually responding to the questionnaire are representative. We do not claim that this study is representative for all universities with a computer science program, but it is useful as an indicator of (a lower boundary of) the state of affairs.

Requests for data was send out to 575 named respondents in November 2006. 78 of the requests for participation were undeliverable, giving a population of 497. Overall 80 respondents (from different institutions) answered the questionnaire, giving a response rate of 16.8%. 17 of the answers were not filled out correctly. Consequently, we have information from 63 institutions only (12,7%). The geographical distribution of responses is presented in Figure 1.
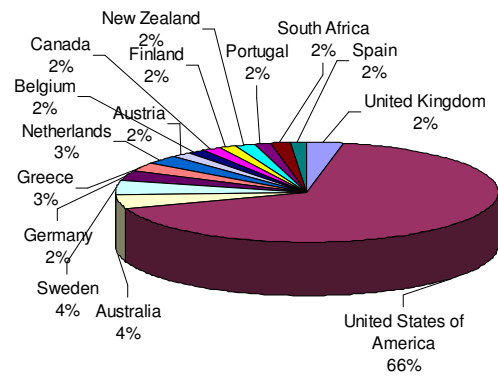

Figure 1: *Geographical distribution of responses*

From Figure 1 it can be seen that the majority of answers originate from the US. We have no access to statistics about the number of students in computer science for each of the countries. The UNESCO Institute of Statistics [16] collects data about education worldwide, but not at a sufficiently detailed level. To give some indication of the distribution of students around the world, UNESCO has numbers for graduates in science in tertiary education (see Figure 2). Unfortunately, UNESCO has no numbers for China, India or other big Asian countries. From UNESCO's distribution of graduates, we conclude that the answers are not as representative as we would have liked.
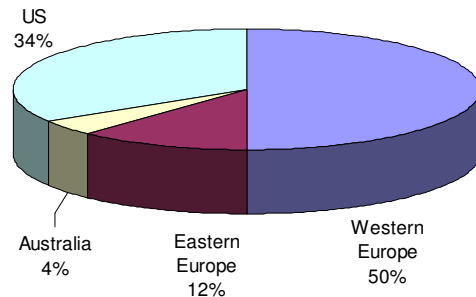

Figure 2: *Percentage of graduates in science in tertiary education according to UNESCO*

3. Results

This paragraph presents the results of our survey.

3.1 Pass, fail, abort, and skip rates

The most interesting question is: "What are the pass and failure rates? —Is it really true that CS1 is a particularly difficult course?" Figure 3 shows that 67% of the students pass (this calculation is based on aggregate numbers, i.e. a course with more students counts more).
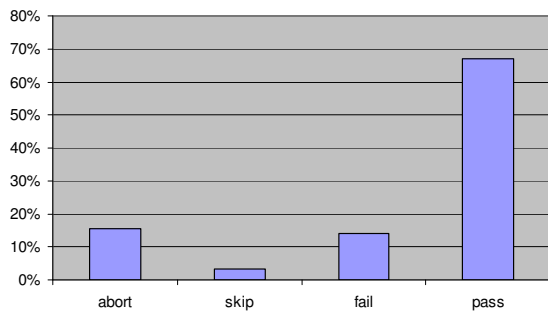
Figure 3: *Pass, fail, abort and skip rates; aggregate*

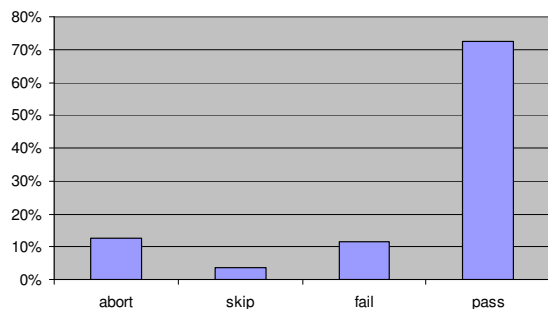Giving all courses equal weight, the result is not changed much; 72 % of all students pass (see Figure 4).

Figure 4: *Pass, fail, abort and skip rates; average*

From these two figures (Figure 3 and Figure 4), it seems difficult to justify the often postulated claim that introductory programming is very difficult and many students fail. (We hypothesize that if we could see the full picture, things would look very different, but we have no data to support this belief.)

There is a huge variation in the pass, fail, abort and skip rates found — from a course with only 5% of the students passing to a course with all students passing (see Table 1 and Figure 5).

	Abort	Skip	Fail	Pass
Mean	0,125	0,035	0,116	0,724
Median	0,108	0,000	0,077	0,706
Standard deviation	0,108	0,098	0,112	0,192

Table 1: *Mean, median and deviation of percentages*

The size of the courses also varies a lot, from 8 students to 645. The mean course size is 116, but 23% of the courses have less than 30 students.

It seems like the small classes (less than 30 students) do better than the larger ones —the average pass rate in the small classes is 82% whereas large classes only have an average pass rate of 69%.
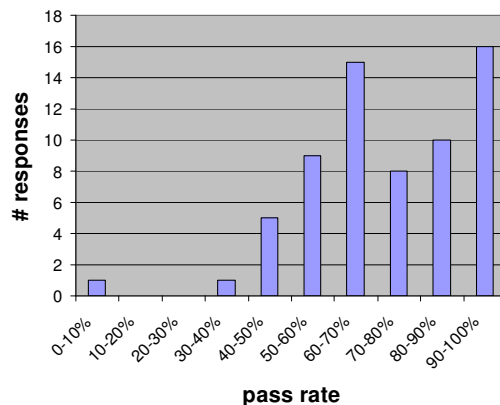

Figure 5: *Pass rate versus number of responses*

3.2 Universities and Colleges

Twelve colleges and fifty universities responded. It seems that the pass rate is higher at colleges than universities; the average pass rate for colleges is 88% whereas it is 66% for universities. However, there are only a small number of colleges so the result is not significant.

3.3 Types of Introductory Courses

Apart from information on pass and failure rates of the introductory courses, we asked for the type of introductory course (object-oriented, imperative or functional). The distribution can be seen in Figure 6.
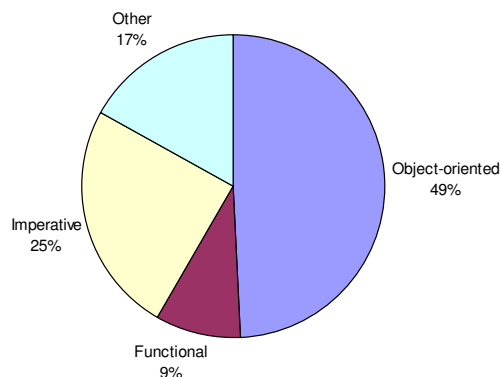

Figure 6: *Distribution of types of courses*

17% report their introductory course to be "other"; almost all of these courses were taught as a combination of imperative and object-oriented.

The pass rate is almost identical for the four types of CS1 courses; thus, this study cannot fuel the ongoing discussion on objects-first versus imperative-first [3].

3.4 The Evaluation of CS1

As part of the questionnaire we asked about the way the students were evaluated. We asked the participants to indicate the weight that each part (final exam, mandatory assignments, etc.) constitutes in percentage of the final grade. There are of course big differences in grading pro-

cedures among the universities. On average, 35% of the final grade is due to marked assignments during the course, 35% is from the final exam, and 30% is from some other source (e.g. lab-exercises, midterm exams, or programming projects).

We have not been able to find any correlation between the way a course is evaluated and the pass-rate.

4. Discussion

4.1 Is the Failure Rate of CS1 High?

The obvious counter question is: what is "high"? Is 33% of the students failing a high number? To relate the number to similar figures, we have looked at the number of students enrolled in tertiary computing education in 1999 and the number of students graduating in 2004 in regions covered by UNESCO. Tertiary studies last from two to eight years; we have decided to use numbers from the years 1999 (enrolment) and 2004 (graduation). For some countries, 1999 enrolment numbers or 2004 graduation numbers were not accessible; in those cases we have used numbers from the neighbouring years [16]. The result is shown in Figure 7. From this figure, the number of students graduating in 2004 was only 26.8% of the number of students enrolled in 1999. In other words, it seems that there is a huge number of students enrolling in tertiary education who do not graduate and in this light, 33% may not be an especially high percentage. (Unfortunately, UNESCO does not have numbers for the US.)
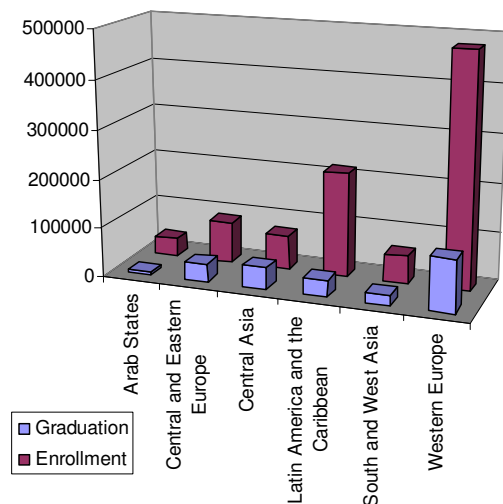

Figure 7: *Enrollment and graduation in computing*

4.2 The Number of Students in Computing

In 1999, approximately one million students enrolled in computing in the 72 countries covered by Figure 7. These 72 countries do not include the US, India and China, so we estimate that more than two million students per year enrol in computing studies worldwide. Assuming that the

pass rate found in this survey is representative, approximately 650,000 students every year do not pass CS1. In this light, just a small improvement of the pass rate of CS1 would cause a gigantic increase in the number of students passing (and perhaps eventually graduating) —a one percent increase in the pass rate means 20,000 students extra passing CS1.

4.3 Quality of Data

Sixty-three institutions provided data for this study. This is a low number; according to *iMahal Resources on Education in the USA* [9], 413 universities and colleges have a program in computer science in the US alone.

A colleague from the ACM Education Council mentioned an internal report of community colleges (two-year schools) in the US who were in a coalition to improve their retention rates in CS. One school reported an average failure rate, over a ten year period, of 90%! And a university with 4000 students, where CS is the second largest major, reported a failure rate of 72%. We have only seen very few similar extreme numbers in our study. One reason could be that teachers and institutions with high failure rates are reluctant to answer this type of questionnaire simply because they are embarrassed by their numbers.

Another source of error could be the selection of respondents; teachers attending and writing about computer science education are likely to be more concerned and proactive in improving their teaching.

5. Conclusion and Future Work

Hard facts about computer science education are highly needed in order to address the most relevant problems. One such fact is the average failure rate or pass rate for different types of courses.

The limitation of this study is the relative low number of respondents. We therefore suggest that the ACM Education Council and others engage in this work in order to provide reliable and representative data from as many institutions as possible.

We did not find the failure-rate of CS1 to be alarmingly high; however, we do not claim that it is possible to make firm general conclusions based on our study.

6. Acknowledgement

7. References

[1] ACEC'04. Proceedings of the Australian Computers in education 2004 conference, Adelaide, Australia July 5-8, 2004.

[2] S. Bergin and R. Reilly. The influence of motivation and comfort-level on learning to program. In Proceedings of the 17th Annual Workshop og the Psychology of Programming Interest Group pages 293-304, University of Sussex, Brighton UK 29 June - 1 July, 2005. University of Sussex,

[3] K. B. Bruce. Controversy on how to teach CS 1: a discussion on the SIGCSE-members mailing list. SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education), 37(2):111-117, 2005.

[4] M. E. Caspersen & J. Bennedsen. Questionnaire for Failure Rates for Introductory Programming Courses. (Last accessed February 9, 2007) http://www.daimi.au.dk/~jbb/questFail.html

[5] G. Engel & E. Roberts. Computing Curricula 2001 Computer Science, Final Report. (Last accessed March 10, 2006) http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf

[6] M. Guzdial and A. Forte. Design process for a non-majors computing course. In SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education pages 361-365, St. Louis, Missouri, USA, 2005.

[7] B. Hanks, C. McDowell, D. Draper and M. Krnjajic. Program quality with pair programming in CS1. In ITiCSE '04: Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education pages 176-180, Leeds, United Kingdom, 2004.

[8] ICALT'04. Proceedings IEEE international conference on Advanced learning technologies. Joensuu, Finland 30 August - 1 September, 2004. IEEE Computer Society,

[9] iMahal. Find Computer Science Colleges and Universities in USA. (Last accessed February 1, 2007) http://www.imahal.com/education/usa/cs/list.htm

[10] ITiCSE'05. Proceedings of the 10th annual conference on innovation and technology in computer science education. Monte de Caparica, Portugal June 27-29, 2005.

[11] T. Jenkins. On the Difficulty of Learning tp Program. In Proceedings for the 3rd Annual conference of the LTSN Centre for Information and Computer Sciences , Loughborough, UK August 27 - 29, 2002.

[12]A. Korhonen and L. Malmi. . Kolin Kolistelut - Koli Calling 2004: Procedings of the fourth Finnish/Baltic Sea Conference on Computer Science Education. Helsinki University of Technology, Department of Computer Science and Technology, Helsinki, Finland, 2004.

[13] S. Lohr. Microsoft, Amid Dwindling Interest, Talks Up Computing as a Career. In http://www.nytimes.com/2004/03/01/technology/01bill.html?ex=1170478800&en=14c1251e099cf4cd&ei=5070 March 1, 2004.

[14] A. Robins, J. Rountree and N. Rountree. Learning and Teaching Programming: A Review and Discussion. Journal of Computer Science Education, 13(2):137-172, 2003.

[15] SIGCSE'05. SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education. St. Louis, Missouri, USA, 2005.

[16] UNESCO. UNESCO Institute of Statistics. (Last accessed January 20, 2007) http://stats.uis.unesco.org/TableViewer/dimView.aspx?ReportId=251