Pass Rates in STEM Disciplines Including Computing

Simon*
University of Newcastle
Australia
simon@newcastle.edu.au

Andrew Luxton-Reilly*
University of Auckland
New Zealand
andrew@cs.auckland.ac.nz

Vangel Ajanovski
Ss Cyril and Methodius University
Republic of North Macedonia
vangel.ajanovski@finki.ukim.mk

Eric Fouh
University of Pennsylvania
USA
efouh@seas.upenn.edu

Chris Gonsalvez
Monash University
Australia
chris.gonsalvez@monash.edu

Juho Leinonen
University of Helsinki
Finland
juho.leinonen@helsinki.fi

Jack Parkinson
University of Glasgow
UK
jack.parkinson@glasgow.ac.uk

Matthew Poole
University of Portsmouth
UK
matthew.poole@port.ac.uk

Neena Thota
University of Massachusetts Amherst
USA
nthota@cs.umass.edu

ABSTRACT

Vast numbers of publications in computing education begin with the premise that programming is hard to learn and hard to teach. Many papers note that failure rates in computing courses, and particularly in introductory programming courses, are higher than their institutions would like. Two highly distinct research projects have established that average success rates in introductory programming courses world-wide are in the region of 67%.

However, there is little published work comparing pass rates in computing courses with those in other STEM disciplines. As institutions continually ask computing educators to justify the atypical failure rates in their courses, a thoroughly researched comparison of this sort could prove useful in demonstrating whether the phenomenon is real, and, if so, whether it extends somewhat beyond the boundaries of individual institutions.

This working group will gather information on pass rates in computing courses, particularly introductory programming courses, and in courses at comparable levels in other STEM disciplines. Members of the group will be required to gather the information from their own institutions, and further data will be gathered by way of a broad survey. The data will be analysed to see whether global patterns can be established, and the group will survey the literature to gather and summarise postulated explanations for any difference between pass rates in computing and in other STEM disciplines.

CCS CONCEPTS

• **Social and professional topics → Computing education**.

KEYWORDS

ITiCSE working group; CS1; introductory programming; pass rate; failure rate; STEM disciplines

ACM Reference Format:
Simon, Andrew Luxton-Reilly, Vangel Ajanovski, Eric Fouh, Chris Gonsalvez, Juho Leinonen, Jack Parkinson, Matthew Poole, and Neena Thota. 2019. Pass Rates in STEM Disciplines Including Computing. In *Innovation and Technology in Computer Science Education (ITiCSE '19), July 15–17, 2019, Aberdeen, Scotland, UK*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3304221.3325532

*Working group co-leader

ITiCSE '19, July 15–17, 2019, Aberdeen, Scotland, UK

ACM ISBN 978-1-4503-6301-3/19/07.
https://doi.org/10.1145/3304221.3325532

1 BACKGROUND

The leaders of this working group were co-leaders of a 2018 ITiCSE working group [6] that conducted a broad-ranging review of the literature pertaining to the introductory programming course. At times it seemed that almost every one of the 1666 papers that they considered had words in its introduction to the effect of 'programming is hard to learn' or 'programming is hard to teach'.

The literature of computing education appears to recognise almost universally that learning to program computers is difficult, and therefore by extension that teaching computer programming is difficult. Most teachers of computing courses accept this, and understand that the pass rates in their courses are typically among the lowest rates at their institution. However, they are often subjected to pressure from their institutions to improve the pass rates, and when that happens they sometimes need to explain that this is a world-wide phenomenon, not specific to the one institution, and it would help if they could point to published research to back up that explanation.

In 2007, using a survey of computing academics worldwide, Bennedsen and Caspersen [1] established an average pass rate of 67% in introductory programming courses. Ten years later, Watson and Li [8] established almost exactly the same pass rate by means of a search of the computing education literature. These rates are not disastrous, but neither are they pleasing to university administrators.

There have been suggestions [2, 3] that some governments are considering tying university funding to pass rates in courses. This

option clearly has the potential to pressure educators into passing students who are inadequately prepared for subsequent courses.

The literature has also seen a number of suggestions as to why pass rates are low in introductory programming courses. For example, Luxton-Reilly [5] suggests that "we make our introductory courses difficult by establishing unrealistic expectations for novice programming students"; Hoda and Andreae [4] suggest that the high level of attrition and failure are due not so much to incapable students as to inadequate teaching; and Parsons et al. [7] suggest that "the methods of assessment ... do not reflect the knowledge and skills that a real programmer needs to write real code".

The first purpose of this working group is to gather data from computing educators worldwide about pass rates in their own introductory courses and pass rates in introductory courses in other STEM disciplines at their own institutions. This will provide an update to the two prior studies of pass rates in introductory computing courses, but additionally it will help to establish whether pass rates in computing courses really are substantially lower than in other STEM courses, and whether this is a universal phenomenon.

The second purpose is to examine the literature for plausible hypotheses about the purported lower pass rates, and, if possible, to synthesise those hypotheses into a viable explanation.

2 METHOD

The working group will conduct a thorough survey of the computing education literature, gathering and summarising plausible explanations of the perceived poor pass rates in introductory programming courses.

Each member of the working group will gather data on pass rates in introductory courses from their own institution, both in computing and in other STEM disciplines. It is expected that each member will gather five years' data. Analysis of this data will give a picture of pass rates at a small number of institutions, and will help to establish whether any trends are discernible over the past five years.

The working group will also conduct a survey in which it asks respondents to provide the same data for just the most recent year at their own institutions. Analysis of that data will strengthen the findings from the institutions of the working group members, giving a clear snapshot of the current relationship between pass rates in introductory programming courses and those in other introductory STEM courses.

Finally, the working group will seek out publicly accessible data on pass rates reported at a state or national level. However, early indications are that such data is not sufficiently detailed to be of any assistance in the working group's endeavours.

REFERENCES

[1] Jens Bennedsen and Michael E Caspersen. 2007. Failure rates in introductory programming. *SIGCSE Bulletin* 39, 2 (June 2007), 32–36. https://doi.org/10.1145/1272848.1272879

[2] Alison Clear, Janet Carter, Amruth Kumar, Cary Laxer, Simon, and Ernesto Cuadros-Vargas. 2015. Global perspectives on assessing educational performance and quality. In *2015 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2015)*. ACM, New York, NY, USA, 326–327. https://doi.org/10.1145/2729094.2754843

[3] Alison Clear and Tony Clear. 2014. Introductory programming and educational performance indicators – a mismatch. In *2014 ITx Conference (ITx 2014)*. CITRENZ, New Zealand, 123–128.

[4] Rashina Hoda and Peter Andreae. 2014. It's not them, it's us! Why computer science fails to impress many first years. In *16th Australasian Computing Education Conference (ACE 2014)*. Australian Computer Society, Inc., Darlinghurst, Australia, 159–162. http://dl.acm.org/citation.cfm?id=2667490.2667509

[5] Andrew Luxton-Reilly. 2016. Learning to program is easy. In *2016 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2016)*. ACM, New York, NY, USA, 284–289. https://doi.org/10.1145/2899415.2899432

[6] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A Becker, Michail Giannakos, Amruth N Kumar, Linda Ott, James Paterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. Introductory programming: a systematic literature review. In *ITiCSE 2018 Working Group Reports (ITiCSE-WGR 2018)*. ACM, New York, NY, USA, 55–106. https://doi.org/10.1145/3293881.3295779

[7] Dale Parsons, Krissi Wood, and Patricia Haden. 2015. What are we doing when we assess programming?. In *17th Australasian Computing Education Conference (ACE 2015)*. ACS, Sydney, Australia, 119–127. http://crpit.com/confpapers/CRPITV160Parsons.pdf

[8] Christopher Watson and Frederick WB Li. 2014. Failure rates in introductory programming revisited. In *2014 Conference on Innovation & Technology in Computer Science Education (ITiCSE 2014)*. ACM, New York, NY, USA, 39–44. https://doi.org/10.1145/2591708.2591749