

An Analysis of Introductory University Programming Courses in the UK

Ellen Murphy
Institute for
Mathematical Innovation
University of Bath
e.murphy@bath.ac.uk

Tom Crick
Dept. of Computing
Cardiff Metropolitan University
tcrick@cardiffmet.ac.uk

James H. Davenport
Dept. of Computer Science
University of Bath
j.h.davenport@bath.ac.uk

ABSTRACT

This paper reports the results of a survey of xx introductory programming courses delivered at UK universities as part of their first year computer science (or similar) degree programmes, conducted in the first half of 2016. Results of this survey are compared with a related survey conducted since 2010 (as well as earlier surveys from 2001 and 2003) on universities in Australia and New Zealand. Trends in student numbers, programming paradigm, programming languages and environment/tools used, as well as the reasons for choice of such are reported. Other aspects of first programming courses such as instructor experience, external delivery of courses and resources given to students are also examined.

The results indicate a trend towards...

Categories and Subject Descriptors

K.3.2 [Computers & Education]: Computer and Information Science Education—*Computer Science Education*;
K.4.1 [Computers And Society]: Public Policy Issues

Keywords

Introductory Programming, Programming Languages, Programming Environments, Computer Science Education, Higher Education, Tertiary Education, UK

1. INTRODUCTION

[8] is the latest in a long line [12, 9] of papers surveying the teaching of introductory programming courses in Australasia. However, such surveys are not the norm elsewhere, and this paper reports the authors' findings from running the first such survey in the United Kingdom.

UK policy context e.g. schools [1, 2], Shadbolt/Wakeham, TEF, graduate employability, etc.

Other work in this space [10, 7, 4, 11, 6]

Also, our previous work [3, 5].

2. METHODOLOGY

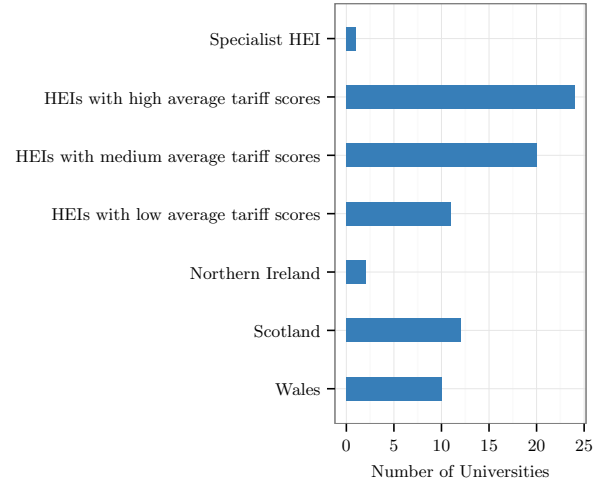


Figure 1: The number of universities per Tariff Group.

Table 1: The number of programming languages used in first programming courses.

Languages	1	2	3	4
Courses	61	17	3	1

2.1 Recruitment of Participants

2.2 Questions

3. RESULTS AND DISCUSSION

3.1 Universities and Courses

3.2 Student Numbers

3.3 Languages

Table 2: The main paradigm in use in the first programming course.

Paradigm	Object-Oriented	Procedural	Functional
Courses	40	27	7

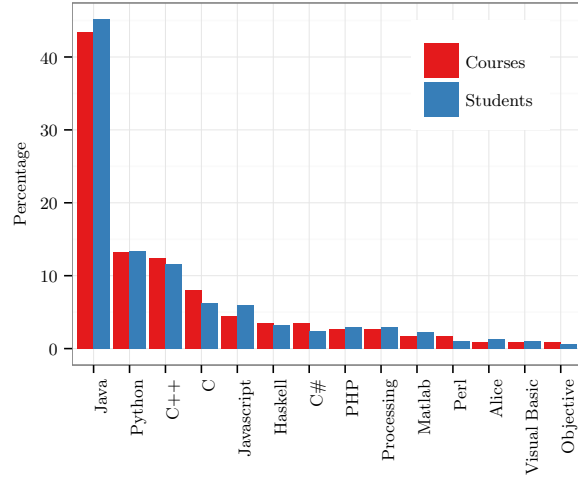


Figure 2: Language popularity by percentage of courses and students.

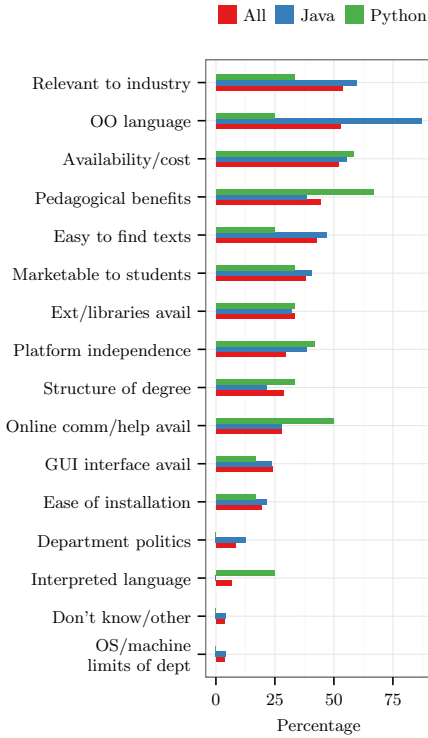


Figure 3: Reasons given for choosing a programming language by percentage for: all languages; Java; and Python.

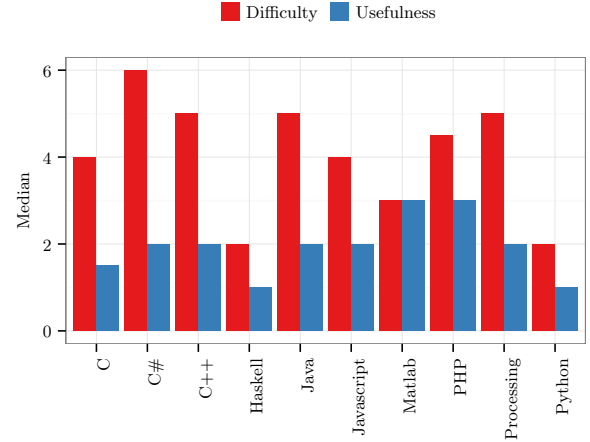


Figure 4: The median of the perceived difficulty and usefulness of language, where 1 is ‘extremely easy’ and 7 is ‘extremely difficult’ for difficulty and 1 is ‘extremely useful’ and 7 is ‘extremely useless’ for usefulness. Answers must have been given by at least two instructors.

Table 3: The number of years the instructor has been involved in teaching introductory programming.

Years	<2	2 - 5	5 - 10	10 - 20	20 - 30	>30
Instructors	3	9	9	27	19	7

3.4 Instructor Experience

3.5 IDEs and Tools

3.6 Other Aspects of the Course

3.7 Aims of an Introductory Programming Course

4. GENERAL DISCUSSION

4.1 The U.K. context

4.2 Comparison with Australasia

Here we compare with [8], the latest Australasian survey.

5. ACKNOWLEDGEMENTS

The authors would like to thank the participants for their engagement with the survey, as well as...

6. REFERENCES

Table 4: The number of tools/environments used in first programming courses.

Tools	1	2	3	4	8
Courses	34	15	6	2	1

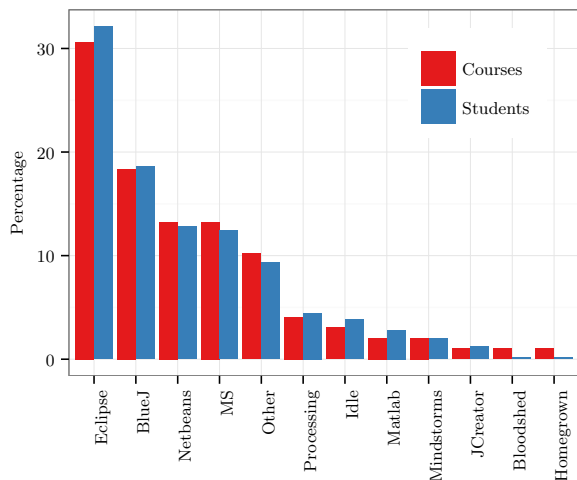


Figure 5: Tool or environment popularity by percentage of courses and students.

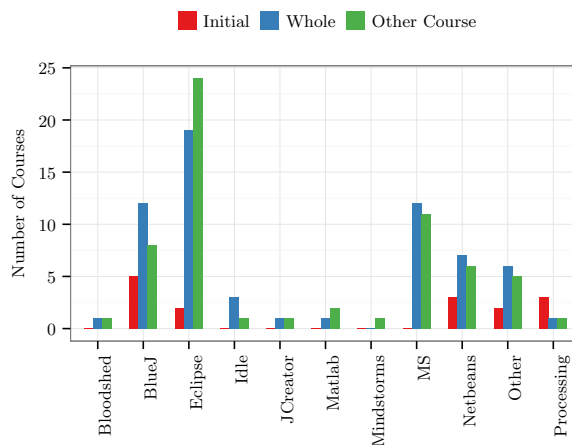


Figure 7: For each tool or environment, whether it is used: for an initial part of the first programming course; throughout the whole of the first programming course; in any other course in the degree.

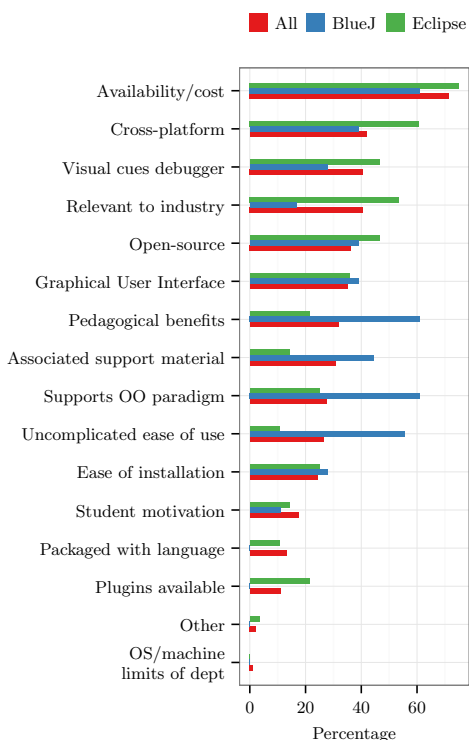


Figure 6: Reasons given for choosing a tool or environment by percentage for: all tools and environments; BlueJ; and Eclipse.

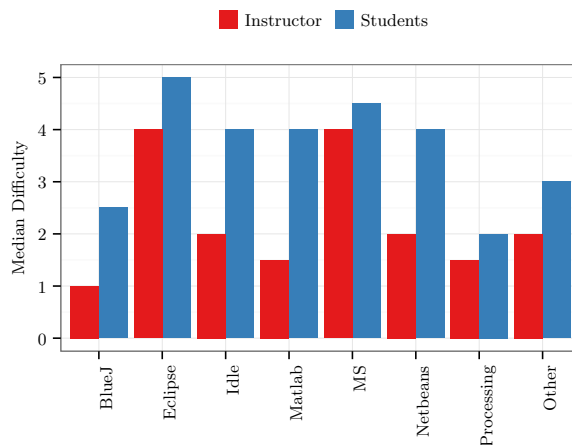


Figure 8: The median difficulty rating of tool/environment for the instructor and students to use, where 1 is 'extremely easy' and 7 is 'extremely difficult'. Answers must have been given by at least two instructors.

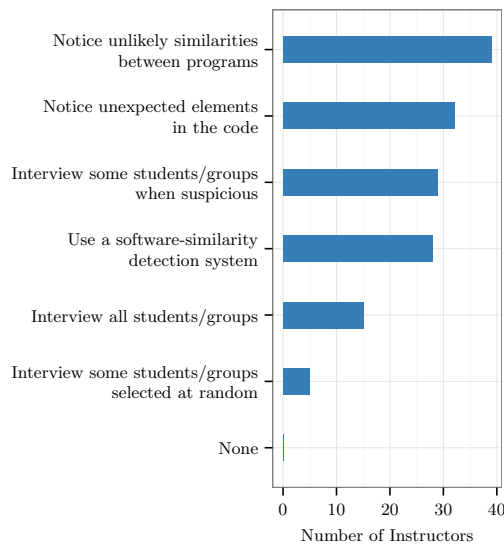


Figure 9: Steps taken to determine whether students have received unauthorised assistance on assignments.



Figure 10: Resources provided to students.

- [1] N. Brown, M. Kölling, T. Crick, S. Peyton Jones, S. Humphreys, and S. Sentance. Bringing Computer Science Back Into Schools: Lessons from the UK. In *Proc. of SIGCSE 2013*, pages 269–274, 2013.
- [2] N. Brown, S. Sentance, T. Crick, and S. Humphreys. Restart: The Resurgence of Computer Science in UK Schools. *ACM TOCE*, 14(2):1–22, 2014.
- [3] T. Crick, J. H. Davenport, and A. Hayes. Innovative Pedagogical Practices in the Craft of Computing. Innovative Pedagogies in the Disciplines. Higher Education Academy, August 2015. <https://www.heacademy.ac.uk/innovative-pedagogical-practices-craft-computing>.
- [4] N. B. Dale. Most difficult topics in CS1: results of an online survey of educators. *ACM SIGCSE Bulletin*, 38(2):49–53, 2006.
- [5] J. H. Davenport, A. Hayes, R. Hourizi, and T. Crick. Innovative Pedagogical Practices in the Craft of Computing. In *Proceedings of 4th International Conference on Learning and Teaching in Computing and Engineering (LaTiCE)*, 2016.
- [6] P. Guo. Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities. <http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities>, July 2014.
- [7] D. Gupta. What is a good first programming language? *Crossroads*, 10(4), 2004.
- [8] R. Mason and G. Cooper. Introductory programming courses in Australia and New Zealand in 2013 – trends and reasons. In *Proceedings of the 16th Australasian Computing Education Conference (ACE’14)*, pages 139–147, 2014.
- [9] R. Mason, G. Cooper, and M. Raadt. Trends in introductory programming courses in Australian universities: languages, environments and pedagogy. In *Proceedings of the 14th Australasian Computing Education Conference (ACE’12)*, pages 33–42, 2012.
- [10] M. McCracken, V. Almstrum, D. Diaz, M. Guzdial, D. Hagan, Y. Kolikant, C. Laxer, L. Thomas, I. Utting, and T. Wilusz. A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, 33(4):125–180, 2001.
- [11] A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin, and J. Paterson. A Survey of Literature on the Teaching of Introductory Programming. *ACM SIGCSE Bulletin*, 39(4):204–223, 2007.
- [12] M. Raadt, R. Watson, and M. Toleman. Trends in introductory programming courses in Australian universities: languages, environments and pedagogy. In *Proceedings of the 6th Australasian Computing Education Conference (ACE’04)*, pages 277–282, 2004.