Teaching Programming in Kenya and South Africa: What is difficult and is it universal?

Jecton Tocho Anyango
Department of Computer Science,
University of Cape Town,
Private Bag X3, Rondebosch 7701
Cape Town, South Africa
anyjec001@myuct.ac.za

Hussein Suleman
Department of Computer Science,
University of Cape Town
Private bag X3, Rondebosch 7701
Cape Town, South Africa
hussein@cs.ac.za

ABSTRACT

Learning to program presents difficulty to most novices at higher education level. In particular, students from poor and middle-income countries may struggle with learning introductory programming. In this paper, we present the issues and difficulties novice programmers face from the lens of lecturers teaching the course from different universities drawn from two African countries.

CCS CONCEPTS

• **Social and professional topics → Computer science education**;

KEYWORDS

Problem solving, Recursion, Introductory programming

ACM Reference Format:
Jecton Tocho Anyango and Hussein Suleman. 2018. Teaching Programming in Kenya and South Africa: What is difficult and is it universal?. In *18th Koli Calling International Conference on Computing Education Research (Koli Calling '18), November 22–25, 2018, Koli, Finland.* ACM, New York, NY, USA, Article 51, 2 pages. https://doi.org/10.1145/3279720.3279744

1 INTRODUCTION

In middle-income and poor countries, it is not necessarily the case that students will have the same educational background and so may learn differently. Supporting this argument, Pillay and Jugoo [7] indicated that students whose first language was the same as the language of instruction (English) performed better in programming than those whose mother tongue was not English in a South African study. People learn differently in different circumstances. However, we cannot predict how students learn when the circumstances are different, therefore we cannot assume that the topics they are going to find difficult to learn are going to be the same in programming. It could therefore be hypothesized that, because of their circumstances, students from middle-income and poor countries would find introductory programming courses far more difficult compared to others from other parts of the world.

Koli Calling '18, November 22–25, 2018, Koli, Finland

ACM ISBN 978-1-4503-6535-2/18/11.
https://doi.org/10.1145/3279720.3279744

Dlodlo [6], through a study of a rural village in South Africa, found that language barrier, shortage of equipment, lack of Internet access, shortage of ICT teachers and experts, inadequate access to hardware and software, and illiteracy were among factors that hindered ICT education for girls and women. Realising the challenges students who have poor background in computational and algorithmic thinking face in learning Computer Science, Barr & Stephenson [2] advocated for the inclusion of these skills in the K-12 syllabus.

There is growing interest in the Computer Science (CS) community to understand the issues and difficult topics for novices. Many works [1] [4] [5] [9] indicated that recursion and parameter passing were difficult topics for most students. There was a consensus in the works of Ambrõsio et al. and Malik & Coldwell-Nelson [1] [5] that recursion was the most difficult and misunderstood concept. Another topic considered difficult was arrays [1] [5] [9].

Though identifying student's misconceptions and difficulties is regarded as a key computer science teacher's competency, relevant research on this topic is not fully developed in the Computer Science Education (CSE) field compared to mathematics and science education [8].

2 AIM

This paper aims at further understanding the topic from the perspectives of lecturers teaching the course from different universities from Kenya and South Africa by answering the following questions:

(1) What do lecturers from middle-income and poor countries perceive as the topmost issue that makes learning introductory programming difficult among novices?
(2) What syllabus content is most difficult for novices and is it the same as for other students from other parts of the world?

3 STUDY DESIGN

3.1 Procedure

Ethical clearance was sought from the Faculty of Science Research Ethics Committee of the University of Cape Town. The purpose of the study was explained to potential lecturers teaching introductory programming courses during the annual Southern African Computer Lecturers'Association (SACLA) 2018 conference. Randomly sampled lecturers were requested to share their contacts and years of experience in teaching the course. The other participants from Kenya were recruited through snowball sampling. In phase one, in-depth interviews were conducted with 5 experienced lecturers who had taught the course for more than 10 years. In

phase two, additional data was collected through an independent online survey with 30 lecturers. Though the sample size could be considered small, the study only targeted lecturers who had taught introductory programming courses in the recent past.

3.2 Online questionnaire design

In one of the previous works [3], a questionnaire was used to identify priority topics for developing a program visualization environment for students. In phase one of the study, one of the items asked the 5 experienced lecturers to mention the problematic issues as well as topics they felt were most critical to teach in the course. These were used to identify the candidate topics and issues to include in the online questionnaire. A summary of some of the topics follows:

(1) Programming basics
(2) If statements
(3) Loops
(4) Arrays
(5) Functions
(6) Recursion
(7) File and exception handling

On the issues making the course challenging and difficult to learn by novices, the following were reported:

(1) Misconceptions about computing
(2) Lack of critical thinking skills
(3) Lack of problem solving skills
(4) In ability to design algorithms
(5) Lack of confidence
(6) Lack of debugging skills

The above topics and programming issues, among others, were considered and refined by the first and second authors who are also experienced lecturers in the course, then used to design an online questionnaire.

4 RESULTS

4.1 Qualitative interview

One of the lecturers had taught introductory programming courses for 14 years, 3 for 13 years while one for 18 years. Open coding was used to organise the interview data to come up with initial categories. Axial coding was applied to establish the relationships between the initial open codes. Further analysis of the axial codes singled out problem solving as the final selective code. Additionally, content analysis revealed that problem solving was the topmost issue with direct phrases from respondents mirroring it representing 48%, followed by algorithm design 20%, student background 14%, teaching approach 10%, and debugging skills 8%. Content analysis of the perceived difficult topic placed recursion on top with 43%, followed by loops 25%, arrays 18% and functions 14%.

4.2 On-line survey

80% of respondents indicated that problem solving was either difficult or very difficult, suggesting it was the biggest problem area. This was followed by 'programming logic', 'debugging'and 'translating logic to code', with 70% for the total of very difficult and difficult in each case. Recursion was perceived as the most difficult topic for novices.

5 CONCLUSION

With results from a survey of 30 and interviews with 5 experienced lecturers teaching introductory programming courses from two African countries, this paper, like others before, [4] [5] [9], confirms that recursion, arrays and abstract data types are difficult for many novices, with recursion being considered the most difficult topic. With regard to the topmost issue making learning programming difficult, results show that problem solving is the issue. It could therefore be argued that despite different learning situations, topics traditionally considered difficult for novices remain the same in middle-income and poor countries as in other parts of the world.

6 LIMITATIONS

The main limitation of the work reported in this paper is that the results are based on surveys and interviews drawn from a moderately small sample. Secondly, it did not consider perspectives from students. It may therefore not infer cause and effect relationships between problem solving and recursion unless a controlled experimental study is conducted with a large sample.

7 ACKNOWLEDGEMENTS

This research was partially funded by the Hasso Plattner Institute, the National Research Foundation of South Africa (Grant numbers: 85470 and 88209) and University of Cape Town. The authors acknowledge that opinions, findings and conclusions or recommendations expressed in this publication are that of the authors, and that the NRF accepts no liability whatsoever in this regard.

REFERENCES

[1] A. P. Ambrosio, F. M. Costa, L. Almeida, A. Franco, and J. Macedo. 2011. Identifying cognitive abilities to improve CS1 outcome. In *2011 Frontiers in Education Conference (FIE)*. F3G–1–F3G–7. https://doi.org/10.1109/FIE.2011.6142824
[2] Valerie Barr and Chris Stephenson. 2011. Bringing Computational Thinking to K-12: What is Involved and What is the Role of the Computer Science Education Community? *ACM Inroads* 2, 1 (Feb. 2011), 48–54. https://doi.org/10.1145/1929887.1929905
[3] Peter Brusilovsky, Jonathan Grady, Michael Spring, and Chul-Hwan Lee. 2006. What Should Be Visualized?: Faculty Perception of Priority Topics for Program Visualization. *SIGCSE Bull.* 38, 2 (June 2006), 44–48. https://doi.org/10.1145/1138403.1138431
[4] Essi Lahtinen, Kirsti Ala-Mutka, and Hannu-Matti Järvinen. 2005. A Study of the Difficulties of Novice Programmers. *SIGCSE Bull.* 37, 3 (June 2005), 14–18. https://doi.org/10.1145/1151954.1067453
[5] Sohail Iqbal Malik and Jo Coldwell-Neilson. 2017. A model for teaching an introductory programming course using ADRI. *Education and Information Technologies* 22, 3 (01 May 2017), 1089–1120. https://doi.org/10.1007/s10639-016-9474-0
[6] Dlodlo Nomusa. 2009. Access to ICT education for girls and women in rural South Africa: A case Study. *Elsevier* 31, 2 (May 2009), 168–175. https://doi.org/10.1016/j.techsoc.2009.03.003
[7] Nelishia Pillay and Vikash R. Jugoo. 2005. An Investigation into Student Characteristics Affecting Novice Programming Performance. *SIGCSE Bull.* 37, 4 (Dec. 2005), 107–110. https://doi.org/10.1145/1113847.1113888
[8] Yizhou Qian and James Lehman. 2017. Students’ Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. *ACM Trans. Comput. Educ.* 18, 1, Article 1 (Oct. 2017), 24 pages. https://doi.org/10.1145/3077618
[9] Gavriel Yarmish and Danny Kopec. 2007. Revisiting Novice Programmer Errors. *SIGCSE Bull.* 39, 2 (June 2007), 131–137. https://doi.org/10.1145/1272848.1272896