International Journal of Online Pedagogy and Course Design
Volume 7 • Issue 1 • January-March 2017

A Pedagogical Review of Programming Education Research:
What Have We Learned

Belle Selene Xia, Department of Information and Computer Science, Aalto University, Aalto, Finland

ABSTRACT

Previous research has shown that especially novice students may find learning programming to be difficult in terms of programming concepts and program design. Many students fail to write correct program codes after the course and achieve the level of skills set by the learning goals of the course. In order to overcome the challenges of learning programming, we are interested in the pedagogical techniques of learning and teaching programming as well as some of the learning tools used to support programming education. This knowledge would help us to better align the design of programming courses with the teaching goal of these courses. New research opportunities are opened in the field of contributing student pedagogy in programming education.

KEYWORDS:

Course Design, Learning Goals, Learning Outcomes, Learning Tools, Pedagogical Research, Program Design, Programming Education

INTRODUCTION

Computing education research is an emergent area of research. Previously teachers have reported their teaching experiences; conferences for sharing these experiences have already emerged in the 1970s; an example is the SIGCSE technical symposium on Computer science education. From approximately the mid of the 1990s until now, researchers in computer science education have rapidly explored the different aspects of teaching in the form of research papers (Fincher and Petre, 2004). Previous research in computer science education has also addressed different aspects of learning including but not limited to learning goals and learning tools used in programming education (Lister and Leaney, 2003; Sondergaard and Mulder, 2012). Research in computer science education contains both evidence and argument. The common methods used in these research papers include both qualitative and quantitative methods. Both types of methods can be used to produce evidence and strengthen arguments. However, researchers are not required to use both types of methods but, instead, may choose to focus on one that is most appropriate for tackling their research questions.

Given the technological advancement, there exists a massive amount of research that details the effectiveness of novel teaching methods and interaction tools used to aid learning programming (Fincher and Petre, 2004). Some of the interaction tools used in programming education to facilitate students' understanding of program execution include program visualization tools, code review tools, algorithm simulation tools and text-editing tools. The text-editing tools used in the integrated development environment, such as Eclipse, are designed to be user-friendly for especially novice

DOI: 10.4018/IJOPCD.2017010103

Copyright © 2017, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

programmers. For them, it is important that the learning tools would expose them to some of the fundamental aspects of the programming process (Kelleher and Pausch, 2005).

Given the importance of learning programming in computer science education, this study aims to open up new insights on some aspects of teaching and learning in programming education as well as the challenges of learning programming in higher education. That is, we aim to detail some of the research results that are relevant to learning programming from especially the novice students' point of view. We have organized our discussions as follows. First, we will go through the background of programming education by discussing the concepts used in this paper. Secondly, we will classify some of the challenges of learning programming and the associated aspects of teaching. Thirdly, we will aim to capture the roles of learning goals and learning outcomes in programming education. Lastly, we will follow up the discussion by going through some examples of learning tools used to support learning programming. Finally, we will summarize and conclude our discussions in the last section of this paper.

1. CONCEPTS AND BACKGROUND OF PROGRAMMING EDUCATION

Bruse et al. (2004) have defined *learning* as widening one's personal experience of a given phenomenon and *teaching* as enhancing students' experience of the given phenomenon through the alignment of critical dimensions in these experiences. According to the theory of constructivism, a teacher may question the student during the teaching process in order to understand what learning models the student possesses, and then attempt to guide them to the correct theories. Constructivism is a theory of learning in which the students construct knowledge rather than merely receiving and storing knowledge shared by the teacher (Mordechai, 1998).

Thompson (2008) has defined learning programming as a process of understanding and applying programming and algorithms knowledge into practice by solving computing problems in an innovative manner. Clear et al. (2009) have classified learning programming according to the level of programming knowledge into the following knowledge levels:

1. Students are unable to write correct code;
2. Students are able to write a small piece of code;
3. Students are able to write a multi-line solution based on a more detailed specification or pseudocode;
4. Students are able to write code to solve a problem which has not been specified to the extent that the problem represents pseudocode for the solution.

In this paper, we define teaching as supporting students to understand the concepts of programming via hands-on experiences and learning as the activity of obtaining useful programming knowledge and skills by studying. In this paper, we aim to discuss the aspects of teaching and learning in programming education as well as the challenges of learning programming in higher education from a pedagogical point of view. This is acknowledged to be a difficult field of research, as research on human thought and behavior including their relationships with learning and computing education studies is not easy (Diamond, 1987).

Soloway (1986) has described programming as a process where an experienced programmer applies schematic knowledge at different levels of abstraction from more complex programming statements or expressions to ones that are less complex. Jeffries et al. (1981) have shown that both novice and expert programmers use a decompositional top-down strategy and forward development to create programs. Specifically, in a top-down design strategy, the programmer approaches the programming problem from a high level of abstraction and decomposes it into subproblems. In forward development, the programmer produces a program in the order in which it appears in the

8 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/a-pedagogical-review-of-programming-education-research/164972

Related Content

"Need to Know": Partnerships in Project-Based Learning
Rhonda Huisman (2014). *Collaborative Models for Librarian and Teacher Partnerships (pp. 121-131).*
www.irma-international.org/chapter/need-to-know/82614

Game-Based Language Learning in Technological Contexts: An Integrated Systematic Review and Bibliometric Analysis
Gwo-Jen Hwang, Pei-Ying Chen, Shih-Ting Chu, Wen-Hua Chuang, Chin-Ya Juanand Hui-Yun Chen (2023). *International Journal of Online Pedagogy and Course Design (pp. 1-25).*
www.irma-international.org/article/game-based-language-learning-in-technological-contexts-an-integrated-systematic-review-and-bibliometric-analysis/316184

Faculty Resistance to Change: An Examination of Motivators and Barriers to Teaching Online in Higher Education
Erin Gratzand Lisa Looney (2020). *International Journal of Online Pedagogy and Course Design (pp. 1-14).*
www.irma-international.org/article/faculty-resistance-to-change/241254

From Course Management Systems to Open Frameworks and Sustainable Ecology of Educational Systems
Vijay Kumarand Jeff Merriman (2005). *Course Management Systems for Learning: Beyond Accidental Pedagogy (pp. 331-343).*
www.irma-international.org/chapter/course-management-systems-open-frameworks/7190

LIS Education ASEANization: An Inchoate Plan?
Reysa Alenzuelaand Ana Mae Bantad Cantel (2020). *Internationalization of Library and Information Science Education in the Asia-Pacific Region (pp. 158-176).*
www.irma-international.org/chapter/lis-education-aseanization/251997