Chapter 74
Assessing the Teaching and Learning Process of an Introductory Programming Course With Bloom's Taxonomy and Assurance of Learning (AOL)

Sohail Iqbal Malik
https://orcid.org/0000-0002-2737-9255
Buraimi University College, Al-Buraimi, Oman

ABSTRACT

Learning to program requires the development of multiple skills including critical thinking, problem-solving, as well as learning the syntax and semantics of the programming language. For novices, to acquire all these skills is considered a challenging and difficult task. They have to focus on both problem-solving strategies and the syntax and semantics of the programming language to acquire these skills. In this study, this article compares the current teaching and learning approach of an introductory programming (IP) course with the six categories of Bloom's taxonomy. The assurance of learning (AOL) process was incorporated in the IP course to assess students' learning outcomes on the basis of achiever (high, medium and low) and performance (very good, good enough and not good enough) categories. The results showed that the current teaching and learning approach of the IP course addressed all the six categories of Bloom's taxonomy. Most of the students (63%) fall under the medium achiever category. Moreover, 50% students learning outcomes come under 'not good enough' performance category.

DOI: 10.4018/978-1-7998-3022-1.ch074

INTRODUCTION

Learning to program is considered as a challenging and difficult task for novice programmers (Malik, 2018a; Malik & Coldwell-Neilson, 2018c; Malik & Coldwell-Neilson, 2017b; Reardon & Tangney 2014; Shuhidan, 2012). Consequently, high attrition rates are reported in introductory programming courses (Guzdial & Soloway, 2002; Lahtinen et al., 2005; Sykes, 2007; Yadin, 2011; Watson & Li, 2014; Zingaro, 2015). Many studies have been performed in the past to determine the challenges and difficulties faced by novice programmers. Tavares et al. (2001) discussed that the two main factors for high failure rates in introductory programming courses are curriculum organization and teaching methods. Meisalo et al. (2002) identified that 30% students' who registered in the programming course dropped out of the course because they found programming exercises too difficult and hard. Iqbal and Harsh (2013) pointed out that introductory programming courses did not pay equal attention to programming knowledge (syntax and semantics) and problem-solving strategies. De Raadt (2008) analyzed forty programming textbooks and determined that only six out of forty textbooks incorporated problem solving strategies throughout the book.

This study compared a current teaching and learning approach used in the introductory programming (IP) course with Bloom's taxonomy and assurance of learning (AOL) process. The comparison with Bloom's taxonomy helps us to determine whether the current teaching and learning approach provides all the necessary skills required by novice programmers in the IP course. The assurance of learning (AOL) process was conducted to assess students learning outcomes and achievements in the IP course.

This paper is organized into a number of sections starting with an introduction to the current teaching and learning approach (ADRI model), Bloom's taxonomy and assurance of learning (AOL), followed by a review of relevant literature. The research questions and methodology used is then described, and results are reported and elaborated. The paper concludes with a summary of the outcomes and future work.

ADRI Approach

The ADRI model is a well-known quality assurance model which is extensively used in education and business sectors (Razvi et al., 2012). It is used by Australian and New-Zealand universities for quality assurance processes.

The current teaching and learning approach in the IP course was based on the four stages of an ADRI approach. Figure 1 depicts the four stages of the ADRI approach.

Malik & Coldwell-Neilson (2016c) explained the four stages of the ADRI approach as shown in Table 1.

Table 2 shows the programming example based on the four stages of the ADRI approach.

The first stage (Approach) of the ADRI model focuses on the problem-solving strategies. A given problem statement is converted into an algorithm by using flowchart and pseudocode techniques. The second stage (Deployment) emphasizes the programming knowledge (syntax and semantics). A computer program is written for the algorithm developed in the first stage by using any programming language. The third stage (Result) covers input, output, and process used to solve the given problem statement. The given problem statement is analyzed, and the required input and output are elaborated. Moreover, the process used to solve the given problem statement is also explained. The fourth stage (Improvement) focuses on introducing new requirements in the given problem statement. The new requirements necessitate programming constructs which are different from programming constructs used in the second stage.