

LEARNING PROGRAMMING USING VISUALISATION – AN ANALYSIS OF LEARNER EXPERIENCES

R.U. Khan¹, F. S. Mohamad², Y.B. Oon², A.S. Shminan², M. H. Barawi²

¹*Universiti Malaysia Sarawak (UNIMAS) (MALAYSIA)*

²*Universiti Malaysia Sarawak (MALAYSIA)*

Abstract

Computational thinking and problem-solving are crucial skills of twenty-first-century education. The abstractness and problem-solving nature of programming is a challenge for novice learners. We experimented with an online free visualisation tool called Python Tutor for Object-Oriented C++ programming to improve the learning of visualisation of abstract concepts, problem-solving and computational thinking. In this study, we engaged two classes of undergraduate students. To investigate the learning effects of the visualisation tool in learning, Class One (C1) was treated as an experimental group, and Class Two (C2) was a control group. The same topics were taught to both groups. The experimental group received an opportunity to use the selected tool as they learned the taught content. At the end of two sessions, a quiz was distributed to both groups. Next, C1 was treated as the control group, and C2 became the experimental group. The same topics were thought to both groups. At the end of the two sessions, a second quiz was given to both groups and scores were recorded. To gauge all participants' problem-solving and computational thinking skills as a whole, we collected data at the beginning and the end of the semester using a Computational Thinking Scales (CTS) and Problem-Solving Test. Findings indicate that the selected visualisation tool helped participants understand and solve ill-defined problems, a critical skill in learning Programming.

Keywords: Programming, Visualisation, Computational thinking, problem-solving skills, novice.

1 INTRODUCTION

Numerous researches have reported a high failure rate in programming courses [1]. Watson and Li [2] observed that the advancement of technology and delivery methods also did not reduce the high failure rate. Watson, Li, and Godwin [2] tried to find out the reasons for this phenomenon, but they concluded that even traditional learning theories could not explain these factors. There have been many attempts to find a relationship between programming behaviour and the high failure rate [3], [4]. The literature on the matter has revealed different interventions used to increase learning programming; unfortunately, it is still unclear why programming is comparatively tricky for most students. It has been observed in many previous studies [5] that most of the students found it hard to pass this subject. Although some interventions have produced good results, there is no clear articulation about the root causes that have affected programming learning.

There are many narratives about why programming is perceived to be a complicated topic. Some believe that programming by nature is complex. Programming is a problem-solving phenomenon. A student needs to build multiple cognitive skills to solve a problem [6] and learn to program [7]. Instead of a complex phenomenon, learning programming also depends on student's internal factors. The student may not want to learn to program, or the student cannot visualise the hidden abstract concept in the program [8], or the student cannot understand the flow of the program [9]. Many students have different passions; therefore, some may lose their motivation [5].

How do these instructional pedagogies affect the passing rates of programming courses? The delivery methodologies or pedagogy of teachers play an essential role. The literature study shows that many teachers have very high expectations while many students fail the subject; it exposes a gap between students' understanding and teacher expectations [10]. Several researchers have highlighted these problems [5]. Many of them had tried to create and apply different interventions to support the students in learning programming [6, 11]. These interventions include a transition from classical lecture-based teaching to different advanced pedagogies such as game-based learning, problem-based learning, or pair programming. Teaching programming is relatively new as compared to other subjects. Therefore its delivery strategies are not well established, and teachers use their styles [12]. Vihavainen, Airaksinen and Watson [2] claimed that no study has quantitatively measured the effect of instructional approaches on reducing the failure rate of programming subjects to date. They found from a systematic review that