International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

IJIEMR Transactions, online available on 13th May 2022.
Link: https://ijiemr.org/downloads/Volume-11/Issue-05

DOI: 10.48047/IJIEMR/V11/I05/16

Title: **MODERN TEACHING METHODS OF PYTHON PROGRAMMING BASICS**

Volume 11, Issue 05, Pages 106-111

Paper Authors. **Ergashev Sirojiddin Gafurovich**

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

MODERN TEACHING METHODS OF PYTHON PROGRAMMING BASICS

Ergashev Sirojiddin Gafurovich

5 general secondary schools of Izbaskan district of Andijan region Teacher of Informatics and Information Technology

ABSTRACT: This paper compares two different approaches of teaching introductory programming by quantitatively analysing the student assessments in a real classroom. The first approach is to emphasise the principles of object-oriented programming and design using Java from the very beginning. The second approach is to first teach the basic programming concepts (loops, branch, and use of libraries) using Python and then move on to oriented programming using Java. Each approach was adopted for one academic year (2008-09 and 2009-10) with first year undergraduate students. Quantitative analysis of the student assessments from the first semester of each year was then carried out. The results of this analysis are presented in this paper. These results suggest that the later approach leads to enhanced learning of introductory programming concepts by students.

Keywords: Teaching programming, object oriented approach, procedural approach, java, python.

In recent years has greatly increased the interest in Computer Science, particularly at secondary schools. In this regard, several countries have carried out thorough investigations of the use of information technology and courses on Computer Science in different educational institutions, which analyses have shown that most of the courses do not meet learners' needs. As a result, several new curricula have been proposed to improve the situation

INTRODUCTION

Teaching programming is a complex task. The task is even more challenging for introductory modules. Failure rates are not marginal (Bennedsen & Caspersen, 2007). One of the common problems shared by Computer Science Departments is the lack of basic programming skills reported by module leaders of courses following first programming courses and how to equip students with better programming skills after the introductory courses. First programming courses typically emphasise the principles of object-oriented programming and design from the very beginning. An alternative approach is based on starting with a more traditional procedural approach first. The evidence from past research seems to suggest that Object Oriented Programming is generally more complex than procedural (Robins et al., 2003). McCane (2009) has demonstrated the effectiveness of Python to teach introductory programming by using qualitative analysis.

In this paper it is therefore suggested to introduce Python (Python Software Foundation, 2010) for the basic (procedural) aspects of programming and then introduce Java (Java, 2010) to focus on object-oriented aspects. The use of Python is expected to both reduce the overhead attached to Java syntax and allow immediate feedback while the students practice with basic instructions (due to the interpreted nature of Python). In other words, at the beginning of the module, novice student are able to devote particular attention to procedural concepts, flow of control, flow of data, etc.

The aim of this paper is to evaluate the success of using Python to introduce basic concepts (loops, branch, and use of libraries). Such evaluation is based on the analysis of student assessments. To measure the success of the proposed method, quantitative analysis of the assessments outcome are presented and discussed. Ehlert (2009) highlights the importance to develop an experimental setting for a fair comparison and the importance to control the variables carefully. The experimental setting used in this paper controls and defines the variables carefully and measures them using clear indicators.

We are social beings, we depend on shared actions, various types of interactions with others. With the current pandemic, a breakdown in the social fabric is associated with several stressors that can negatively affect people's mental health. In this situation, people have changed their behaviors by adopting social distancing measures. Indeed, humanity is threatened by a global pandemic. The latter can last as long as a vaccine or an effective medicine is not found. Vision is not clear and the stress caused by this pandemic is releasing "negative" chemicals into our brains. Furthermore, it is more than necessary to produce beneficial activities to affect the reward zone in our brain. In this context, school is a real environment of beneficial actions based on human interactions.

However, the emergence of Covid-19 has disrupted all facets of daily life around the world. Everywhere around the world, Morocco is no exception, education was one of the sectors that were harshly hit by the closure of schools: the distance is needed as the only operational solution to save the current situation. As a result, education in Morocco is experiencing a real challenge due to the consequences of covid19. This challenge has materialized in the transformation of the face-to-face, presential, teaching mode into distance learning.

Additionally, there has been little pedagogical consistency in how this change has occurred. Months later, the national education ministry adopted "Blended Learning" in all educational cycles when the schools partially reopened. This emergency plan caused several problems for learners, teachers and policy makers. Indeed, a change of this magnitude requires technological prerequisites for learners and techno-pedagogical prerequisites for teachers.

The following research will be focusing on teaching computer science, and more specifically programming in python. The choice of python is not arbitrary. In the programming world, Python is increasingly the most answered, as well as, the most popular. However, students are not interested in pursuing their studies in this field. The sources of this problem are diverse. Students encounter psychological, epistemological and didactic problems.

As part of this research, the didactic obstacles will also be explained. The existence of several pedagogical approaches to teaching python is an advantage that cannot be neglected. Nevertheless, this diversity makes the teacher's job progressively complicated. Indeed, each teaching approach has its advantages and disadvantages.

According to computer educators, the investigative approach, the problem-solving approach, the trial / error strategy will only be beneficial during the construction of programming knowledge. The challenge is to take advantage of these approaches to promote an effective and upbringing learning; That is why a "handy" solution was adopted for teachers in the form of an educational scenario. The following work is based on the scenario that integrates the most appropriate pedagogical approaches for teaching and learning programming concepts. Also, the scenario takes into consideration the importance of group work (the collaborative dimension) in addition to the constraints generated by the pandemic and social distancing (blended learning).

Various approaches to teach programming have been summarised in (Lemos, 1979) and (Engel et al., 2001). But there is an ongoing debate in the teaching community over the best approach to teach introductory programming (Lister et al., 2006; Pears et al., 2007). We have found four experimental studies that compare object oriented approach with the traditional procedural approach. The first study by Decker (2003) has found no difference in student performance between the two approaches. The second study by Reges (2006) has found significant gains in student satisfaction and enrolment after replacing the object oriented programming first curriculum with a procedural approach. The third study by Vilner (2007) has found no significant gains in student performance between the two approaches. The parameters used in this research to compare the two teaching approaches are the pass rate of students and grades in questions related to recursion, efficiency of algorithms and designing of classes. The fourth study by Ehlert (2009) has found no significant gains by changing the object-oriented programming first approach with object oriented programming later approach.

According to Ehlert (2009), different pegagogical dimensions complicate the analysis of different approaches to teach programming. Bruce (2005) argues that there is a need for more experimental studies to examine the different approaches to teach programming.

This paper presents experimental results from real classroom comparing two different approaches to teach introductory programming. The first approach is to emphasise the principles of object-oriented programming and design using Java from the very beginning. The second approach is to first teach the basic programming concepts (loops, branch, and use of libraries) using Python and then move on to oriented programming using Java.

The educational approaches are diverse. In the literature field, the approach is represented by objectives, skills, problem solving, project and, investigation. Each has its own advantages. Teachers of programming generally and python particularly are invited to find the most appropriate recipe to transmit and build new knowledge and skills to the learners. From these approaches follow a set of methods and techniques to promote learning. Unfortunately, there is neither an obvious solution nor a magical want to make learning more effective and sustainable. However, experiments and methods have given promising clues. The study and analysis of these experiences will only be beneficial in helping to improve the current situation. Some authors suggest research to test different pedagogical approaches. Yet, the diversity of educational aspects complicates the analysis of the different approaches of the educational programming. To illustrate, Bruce argues that there is a need for additional and experimental research to explore the different pedagogical approaches of the teaching program.

Researchers suggest using "feedback" to obtain a better result. This strategy focuses on learners' errors, in order to build their scientific knowledge. Still, this method, based on the pedagogy of error, is quite painful for the teachers as it contains elements of individual learning that are costly in terms of time.

Other researchers argue that a reform is needed in the teaching of the Python programming language: The methods of training and assessment of the examination system are an urgent problem of the proposed reform. According to Begosso, Python is designed to optimize the productivity of programmers by offering high-level tools and an easy-to-use syntax. The resource richness of the open community also poses new challenges, large amount of information will require a lot of attention. Kui propose a "Mode of teaching by visualization: visualization teaching mode". By dynamically adjusting the curriculum, scientifically planning educational content, and effectively using appropriate educational approaches, the quality of teaching can be improved From another perspective based on the learning pyramid and retention rate, Fagan and Payne consider that the best way to learn python is to teach it.

The first teaching approach using Java has been followed for the academic year 2008-09 and the second teaching approach using Python has been followed for the academic year 2009-10. The student assessments towards the end of first semester in each academic year have been used to carry out the experiments described in this paper. Three indicators namely grades, programs with bugs and frequency of keywords have been used for comparison. The grades of the submitted program are a measure of success of the student ability to implement programs. Bugs are interpreted as a measure of their overall understanding of programming. Frequency values of keywords are interpreted as a measure of the familiarity with basic concepts of programming. The four keywords used in the frequency measure are “if”, “for”, “while” and “import” (in combination with the use of “random” class).

The experimental results discussed in this paper show a measured positive (increase) in all the three indicators in favour of the second approach which is to first teach the basic programming concepts (loops, branch, and use of libraries) using Python and then move on to oriented programming using Java. The use of Python could facilitate the mastering of basic concepts such as loops, branch, and use of libraries for novice students. A possible explanation could be that with Python students can focus on the crucial basic issues without being distracted by the overheads. Moreover, the complexity of the programs used during their practice could be tailored in a more accurate way to their level of proficiency.

One limitation of this research is that it considers only one case study. Such case studies require considerable amount of time as they have to fit in with the academic calendar and hence we could only consider one case study. We hope that similar case studies will be replicated by other institutions as well. The results from our case study are positive and statistically significant in favour of our proposed approach which is to first teach the basic programming concepts (loops, branch, and use of libraries) using Python and then move on to oriented programming using Java. We hope that our research will encourage debate over different approaches to teaching introductory programming and in turn will lead to adoption of better teaching methodologies. Also this research focuses only on the quantitative analysis of the student assessments and does not elicit the views of the students about their learning experience. So an interesting direction for future work would be to collect and analyse the qualitative data from the students about their perspective. This research takes into account the performance of students up until the first semester. We are analysing the performance of students over the whole academic year and will be publishing it at a later stage.

REFERENCES

1. Bennedsen, J., Caspersen, M.E. (2007), Failure rates in introductory programming. ACM SIGCSE Bulletin, 39(2), pp. 32–36.
2. BlueJ, (2010), BlueJ - Teaching Java - Learning Java, http://www.bluej.org/ (01 Dec 2010)
3. Bruce, K.B. (2005), Controversy on how to teach CS 1: a discussion on the SIGCSE-members mailing list, ACM SIGCSE Bulletin, 37(2), pp. 111-117
4. Decker, A. (2003), A tale of two paradigms, Journal of Computing Sciences in Colleges, 19(2), pp. 238-246
5. Engel, G., Roberts, E. (2001), Computing Curricula 2001 Computer Science: final report, IEEE Computer Society and Association for Computing Machinery, 28, pp. 2004
6. Ehlert, A., Schulte, C. (2009), Empirical comparison of objects-first and objects-later, ACM Fifth international workshop on Computing education research workshop, pp. 15-26
7. Java, (2010), Oracle Java Technologies, http://www.oracle.com/us/technologies/java/index.html (01 Dec 2010)
8. Lemos, R.S. (1979), Teaching programming languages: A survey of approaches, ACM SIGCSE Bulletin, 11(1), pp. 174-181