European Proceedings of
Social and Behavioural Sciences
EpSBS

www.europeanproceedings.com

e-ISSN: 2357-1330

DOI: 10.15405/epsbs.2022.11.55

SCTCMG 2022

International Scientific Conference «Social and Cultural Transformations in the Context of Modern Globalism»

METHODS FOR LEARNING PROGRAMMING

Janet Movladievna Magamedova (a)*, Makka Saidakhmetovna Turpalova (b),
Zarina Mukhmadovna Lorsanova (c)
*Corresponding author

(a) Chechen State Pedagogical University, Chechen State University A.A. Kadyrov
(b) Chechen State Pedagogical University, Chechen State University A.A. Kadyrov
(c) Chechen State Pedagogical University, Chechen State University A.A. Kadyrov

Abstract

Pedagogical interaction in the "teacher – university student" system provides a system of reciprocal influences of the subjects involved in the common work on the basis of the aggregate goals of vocational education. Collaborative work between a teacher and a student influences the formation of a future specialist's system of values related to personal attitudes, truth, education, profession and other. The effectiveness of pedagogical interaction depends on a large number of subjective moments, and the most significant role belongs to the model of rational choice of teaching methods, the implementation of which in the criteria determined by the educational institution realizes the task of forming a high-class specialist-graduate of an educational institution. Modern society is inseparable from information technology, and as a result, the list of available teaching methods is replenished with innovative methods that facilitate both teaching for teachers and the perception of information for students. However, the flow of incoming information is so wide and varied that it is quite difficult to isolate sometimes the necessary material and work only with it, without being distracted by other topics. Under such conditions, the teacher needs a certain skill to capture and hold the attention of students throughout the entire lesson. Effective teaching methods and motivational policies pursued by the teacher in their discipline can come to the rescue. This work is presented as a research experiment on the methodology of conducting practical programming lessons.

2357-1330 © 2022 Published by European Publisher.

Keywords: Education, experiment, interactive methods, programming, teaching methods

The Author(s) 2022. This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License (https://creativecommons.org/licenses/by-nc-nd/4.0/).

https://doi.org/10.15405/epsbs.2022.11.55
Corresponding Author: Janet Movladievna Magamedova
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

1. Introduction

The end goal of an educational interaction between a teacher and students is formation of a system of values in future specialists; these values include the human values, truth, education, profession, etc. It is important to bear in mind that the stated interaction between the teacher and the students are not always calm (Lisitsin, 2010). There may be conflicts of goals, interests, attitudes, motives, etc. during this process. In order to overcome certain conflict situations or resolve arising problems, tertiary education sees successful implementation of interactive methods of learning to supplement those previously used. Interactive learning is a learning built upon interaction between all the students and their instructor. But instructor's activity shall yield to learners' activity, so that learners are always involved in the process of cognition and knowledge exchange (Matiz, 2017).

2. Problem Statement

During practical training, instructors works with twenty or more students. More often than not, the ability level of the students is quite varied and so during the practical training it is important that those more capable do not hold all the attention of the instructor. If such a situation arises, those less capable may close up and lose their interest in the subject. In order to mitigate this situation, the work may be organized in pairs of students (Gileva et al., 2018). However, history of education knows various models of teaching, which in one form or another are used in educational institution. For example, interaction between instructor and students may be organized following a scheme where the instructor serves as a moderator. In this case, their principal goal is to activate student activity by organizing discussions between the students aimed at solving some existing problems. When explaining some topics, the instructor may serve as a consultant. This model is suitable when it is necessary to teach students to independently find solution for a specific given task, that is, in arrangement of student independent work (Lutz, 2019). When a task is issued to a group of student, the instructor takes up a role of a tutor. The instructor may serve as a trainer when working with more capable students.

3. Research Questions

Let us consider some methods as exemplified in teaching programming to first-year students. Programming languages in educational institutions are usually taught for two or more terms. Parallel studying of several programming languages may introduce certain confusion in learners' knowledge and undermine their self-confidence. While an experienced programmer is capable of taking what he knows about cycles and function calls in one language and reuse this understanding in a language with a different syntax, a novice is lacking the understanding of which elements of their knowledge are central and which are random. An optimal solution to the problem may be a variant of a tutoring instructor, when some learners obtain additional knowledge by being instructed by their peers. Essentially, it is an example of a large-scale individual mentorship within an educational institution. Group discussion significantly improves understanding of studied material as it prompts students to make their cognition clearer, which may be just enough to reveal gaps in reasoning.

https://doi.org/10.15405/epsbs.2022.11.55
Corresponding Author: Janet Movladievna Magamedova
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

4. Purpose of the Study

During practical training, the instructor has an ability to present a new topic as a presentation (e.g., in PowerPoint). However, if the material contains executable code, demonstrating operation of a compiler in front of students is more effective, as it creates a number of advantages.

1.It allows for more active reaction to what-if questions.

While slides leave no place for experiments, working with a software development environment in front of audience reminds of sailing in unchartered waters, as a situation may arise where some code is not running or results in an error following unexpected questions from students. Nevertheless, it is what may be called live programming and it is much more interesting to students.

2.It facilitates unintended transfer of knowledge, when students are getting to know more that their instructor was intended to pass along.

3.Conclusions on behalf of the students in the vein of "programming is not so scary", "errors can be corrected", "I also may become an author of a program", etc.

5. Research Methods

For instance, quite possible is a situation where *executable code produces an error in case of live programming*, as the instructor could introduce a mistake while typing. Most students model their behavior after that of their instructors, usually without realizing it. Maybe not all instructors are capable of following this model, however, if they are not afraid of making mistakes and discussing them in front of the class, the students will be interested in new traditions and possibilities to experiment while completing tasks without hesitations in their answers (Bukunov & Bukunova, 2017).

6. Findings

Explanations of a new topic may employ images, audio and video materials demonstrating implementation of algorithms in a specific programming language. With a high possibility, in this case the instructor will be capable of holding students' attention and it will facilitate better retention of material. Students find real-life problems more engaging than abstract tasks.

An instructor may present their subject as a large-scale project and appear as a manager of the project, trying to get student interest (Sweigart, 2016). The more successful is such a presentation, the higher is the probability that all the competences actualized in the subject will be mastered by the students. In any case, the main point is student motivation with the future prospects of their selected profession in mind.

When an instructor issues a task to students, those more into the subject are starting to complete the task using individual methods of solution (Figure 1–4). For instance, an instructor issued a beginner-level group the following task: "Write a program that calculates a difference between the number and the sum of its digits".

https://doi.org/10.15405/epsbs.2022.11.55
Corresponding Author: Janet Movladievna Magamedova
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

summa_.py - C:\Users\lenovo pc\Desktop\summa_.py (3.9.7)

File Edit Format Run Options Window Help

```
number = int(input("Введите целое: "))
def F(number):
    sum = 0
    while (number != 0):
        sum += number % 10
        number//= 10
    return sum
print(number-F(number))
```

Figure 1. Solution produced by the main group together with the instructor

Hamiev.py - C:\Users\lenovo pc\Desktop\Han

File Edit Format Run Options Window He

```
def A(kar):
    summa=0
    for i in kar:
        summa+=int(i)
    razn=int(kar)-summa
    print(razn)
A(kar=str(input()))
```

Figure 2. Student solution

Yanduev.py - C:\Users\lenovo pc\Desktop\Yandu

File Edit Format Run Options Window Help

```
a=input()
sum=0
for i in range(0,len(a)):
    sum+=int(a[i])
print(int(a)-sum)
```

Figure 3. Student solution

Musaev.py - C:\Users\lenovo pc\Desktop\Musaev.py (3.9.7)

File Edit Format Run Options Window Help

```
def func(num):
    sum=0
    for i in num:
        if i.isdigit():
            sum+=int(i)
    x=int(num)-sum
    print("\n","Ответ:",x)
func(str(input("ввести число:
```

Figure 4. Student solution

https://doi.org/10.15405/epsbs.2022.11.55
Corresponding Author: Janet Movladievna Magamedova
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

Many instructors employ computer games as motivational examples in programming classes. One of the variants is learning programming online through gaming trainers. Whatever are the examples, the goal shall be to make a quicker transition from complex and boring algorithms to easy and captivating ones. However, inexperienced programmers code differently than experts do and they need different approaches or tools. For example, if there is a need to search through an array of real numbers to get the minimum value, one may write a short code, using the knowledge about a required pattern (existence of the **min** function, Figure 5).

```
from array import*
arr = array('f', [3.7, 5.1, 1.8, 0.12,0.87,9.054])
print("{:.2f}".format(min(arr)))
```

Figure 5. Function **min** in Python

Inexperienced programmers may approach this problem in different ways, however, most of them will use their available skills and start a process of searching through array elements using a parametrized cycle operator (Figure 6).

```
from array import*
arr = array('f', [3.7, 5.1, 1.8, 0.12,0.87,9.054])
min=arr[0]
for i in range(1,len(arr)):
    if min>arr[i]:
        min=arr[i]
print("{:.2f}".format(min))
```

Figure 6. Finding the minimum element using a cycle

In order to help students attain visible and satisfactory result, instructors may present several pre-written libraries or pieces of starting code that bring students closer to their objective.

The students shall be reminded that despite advertisement campaigns telling them that one may become a professional software developer fast, any job requires experience. Novice programmers will need to gradually learn the foundations of the subject. Possibly it would require spending some time laying out the algorithm on paper as step-by-step instructions (experienced programmers usually skip this stage, having already memorized the most common algorithmic patterns). At the early stage of learning, there may be issues with program debugging (Richter, 2016). It is just natural. So, when children are learning to read they do not start by reading mysteries and romance novels, they start by reading small books with simple sentences in large print. More serious books are read when the reader has mastered the skill of reading and already has a certain breadth of knowledge. Similarly, there is a set of sources for novice programmers, mastering which allows them to advance to the next level, where programming algorithms are more complex. Programmer education in a tertiary school shall teach students practical skills needed in their practical work, skills in using git, code writing discipline, full use of IDE capabilities, etc. The objective of higher education is to provide fundamental knowledge. However, it

https://doi.org/10.15405/epsbs.2022.11.55
Corresponding Author: Janet Movladievna Magamedova
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

shall be taken into account that some programming languages and frameworks may be updated or become obsolete, and as a rule, instructors are required to regularly update materials used in lectures and practical training. For example, 20 years ago, C++ was thought of as a foundational programming language. Examples in this language were used to illustrate memory management, OOP, creating a class hierarchy, UI design, etc. However, frequent updates and extensions complicated work with C++ and despite expansion of capabilities of the language, more flexible languages that appeared later became prevalent. All these languages were slower than C++, but in the age of accelerated development of computer hardware it is not critical.

At the first stage of teaching programming, it is possible to use Python–an interpreted language that runs the program on a virtual machine. There are many additional components (packages) available for various Python application areas. Such packages may be installed on or accessed by the virtual machine. At that, every Python program is in and of itself a package. In any case, instructor shall not as much provide knowledge to students as instill them with a skill of continuous independent learning, regardless of the programming language being studied. Another principle that shall be voiced is "the moment you relax, stop looking for new things and progressing is the moment your knowledge starts becoming outdated". This principle is topical not just to students, but to any modern instructor regardless of their experience and employment history.

7. Conclusion

In the end, one may say that the value of higher education lies in understanding of cause-and-effect relations (why certain approaches were formed and not some others), understanding that any tool, method or concept have limits to their applicability, acquiring a skill of abstract concept thinking. This research allows concluding that application of alternative methods of learning in teaching programming may become an efficient tool for attaining set learning goals.

References

Bukunov, S. V., & Bukunova, O. V. (2017). *Fundamentals of object-oriented programming.* St. Petersburg State University of Architecture and Civil Engineering; EBS DIA. http://www.iprbookshop.ru/74339.html

Gileva, A. V., Gilev, Y. Y., & Richter, T. V. (2018). *Active and interactive teaching methods in natural and mathematical education.* Compiled by T.V. Richter. Solikamsk State Pedagogical Institute. https://www.iprbookshop.ru/86551.html

Lisitsin, D. V. (2010). *Object-oriented programming*. Novosibirsk State Technical University. http://www.iprbookshop.ru/44970.html

Lutz, M. (2019). *Learning Python*. Dialectika LLC.

Matiz, E. (2017). *Learning Python. Game programming, data visualization, web applications*. Peter.

Richter, T. V. (2016). *The use of interactive teaching methods in the educational process of higher education in the formation of students' professional competencies: study guide*. Solikamsk State Pedagogical Institute. https://www.iprbookshop.ru/86544.html

Sweigart, A. (2016). *Automating routine tasks with Python: A practical guide for beginners*. Publishing house "Dialectics / Williams".