Cypriot Journal of Educational Sciences

Volume 15 Issue 5 (2020) 1238-1248

www.cjes.eu

The effect of applying 4-stages on learning analysis and design of algorithms

Ezgi Deniz Ülker *[a,] Department of Computer Engineering, Faculty of Engineering, European University of Lefke, Mersin-10, Turkey https://orcid.org/0000-0002-9642-7310

Suggested Citation:

Ülker, E. D. (2020). The effect of applying 4-stages on learning analysis and design of algorithms. *Cypriot Journal of Educational Science*. *15*(5), 1238-1248 https://doi.org/10.18844/cjes.v15i5.4621

Received from August 15, 2020; revised from August 25, 2020 ; accepted from October 25, 2020.

Selection and peer review under responsibility of Prof. Dr. Huseyin Uzunboylu, Higher Education Planning, Supervision, Accreditation and Coordination Board, Cyprus.

Abstract

The ability of analysing and designing an algorithm is quite essential for computer science education. The students in the Analysis and Design of Algorithms (ADA) course are expected to be able to solve problems by choosing one of the proper design methods and to analyse the algorithm's performance in terms of various aspects. Instead of using traditional teaching approaches, a new 4-stage method is proposed to improve the concept of learning and teaching of algorithms. The instructor applied 4 consecutive stages during the tutorials and practical implementations. The students were asked a series of questions for evaluating not only themselves but also the new method and the instructor capability of handling the course by applying the 4-stages. According to the evaluations, the 4-stage method met the course objectives with noticeable rates and the course learning outcomes were achieved with high success rates when compared with the previous semester. In order to attain a sustainable achievement in teaching ADA, it is recommended to apply the 4-stage method every academic year. The application of 4-stages allows the instructor to be more efficient in programming teaching and the student more confident in programming, therefore it can be suggested for other programming related courses.

Keywords: Computer science education; Concepts of learning and teaching; Course learning outcomes; Programming teaching; Teaching approaches.

* * ADDRESS FOR CORRESPONDENCE: Ezgi Deniz Ülker, Department of Computer Engineering, Faculty of Engineering, European University of Lefke, Mersin-10, Turkey

E-mail address: eulker@eul.edu.tr

1. INTRODUCTION

Over the years, due to increasing development in computer science, programming courses have become interesting and particularly analysing and designing of the algorithms have gained more importance. Computational programming is not only for computer engineering or software engineering students. It is preferred by instructors in teaching of some related fields such as electromagnetism (Nogueira, Alves & Marques, 2019), power system measurement (Lin, 2006) and science (Hamzat, Bello & Abimbola, 2017). Teaching analysis and design of algorithms (ADA) to the second-year students has been a difficult task for the instructors. The content of the ADA course builds the fundamentals of almost all computer science courses and for this reason, it is crucial for the students to learn the concepts in detail. However, the students generally encounter difficulties in this learning process and the instructors have always been seeking for new methodologies in the teaching of programming (Begosso et al., 2012).

An algorithm is an expression of the requirements to solve a class of problems. During the algorithm design, one of the main concerns is to find the best problem solving technique to accomplish the task efficiently in terms of time complexity. The efficiency of the algorithm is strongly related to the technique used (Levitin, 2008). The problem solving techniques; analysing and designing, are considered as continuous processes, until a better solution is obtained. These are also recognized as preliminary steps before implementing an algorithm. Because of this, the students are required to comprehend the steps of analysing and designing at the end of the course and to be able to choose appropriate design techniques for a given problem such as the divide and conquer technique, greedy algorithm techniques, backtracking method and dynamic programming. Considering the difficulties of teaching and learning the ADA course, a new teaching strategy is proposed in this paper. This new methodology is called the 4-stage method and it is based on four different and consecutive steps. Main motivations in proposing the 4-stage method can be listed as follows:

- Despite the inspiring course objectives, the students do not have enthusiasm for this course.
- The instructors have discovered that they spend lots of time and effort teaching, but still most of the students are not satisfied.
- Most of the students agreed on the difficulty of the programming courses and there is a high rate of failure or drop out.
- Learning outcomes achievement rates are low according to the collected feedback.

According to Du Boulay (1986), there are five strategies that must be dealt with in teaching:

1. Orientation; why the programs are needed and what they are used for.
2. Notional machine; essentials of computers to run the programs.
3. Notation; syntax and semantics of programming language which is used.
4. Structures; schematic structures/algorithms of the program.
5. Pragmatics; skills of expert programmers such as planning, developing, testing etc.

The students generally do not have any background with these five strategies. Additionally, they do not have much interest to solve the problems. Also, three competencies that are needed for improvement in the course are scientific competence, algorithmic thinking and efficiency in independent learning. In order to attain the teaching strategies outlined by Du Boulay (1986), and improving the competencies, the 4-stage method is included in the teaching process.

According to Dreyfus & Dreyfus (1986), there are five types of students in a programming class; novice, advanced beginner, competence, proficiency and expert. The 4-stage method is structured to accommodate all five types of students. It contains four consecutive stages in teaching analysis and design of the algorithms in depth. As Du Boulay (1986), expresses in the five strategies of teaching programming, we follow the similar procedure in teaching the ADA course.

The first stage relies on understanding the concept and targets for scientific competency. The concept of each topic needs to be understood clearly by the students. The instructor is responsible to develop a conceptual knowledge related to the program understanding or structuring basic syntax and semantics of programs, before any program is implemented by the students. It is declared that the instructor's behaviour during the tutorial classes is directly related with the problem solving skills of the students in programming (Hooshyar et al., 2016; Yıldız, Alkan & Cengel, 2019). Therefore, this stage aims to reassure problem solving skills of the students.

The second stage encourages the students to form an algorithm using any programming languages. In a way, algorithmic thinking competency is targeted in this stage which requires deep problem understanding in a tutorial class and aims to give experience to the students in making choices of selecting the efficient design techniques for the problems.

The third stage aims to give competency to students in terms of independent learning and it is based on the different learning levels of the students. It is quite important to give various kinds of examples to keep the students' interests alive during learning. We believe that a programmer must analyse and design his own algorithm regardless of his ability of programming. Designs can contain bugs or can be incorrect, but even the novice programmers should be able to track for understanding the program's behaviour (Perkins et al., 1989).

The fourth stage is only recommended after students have gained enough experience and confidence in analysing and designing of algorithms. In this stage, time duration is given to the students and expected to build the program within this duration. According to Bonar & Soloway (1989), developing a program is a step-by-step process and when one step is missing or not comprehended clearly, the program concludes with errors. Under the light of this statement, we do not recommend to apply stage 4, unless the first three stages are applied consecutively.

The students generally have less attention to analyse and design algorithms because they find the problems complex or unexciting (Winslow, 1996). Every student has a unique experience in the learning process, but the teaching methods are considered as if all of the students' learning rates are the same (Begosso et al., 2012). Due to the unsatisfactory learning rates on programming courses, the instructors search for new teaching strategies instead of applying traditional teaching techniques (Milne & Rowe, 2002).

According to Winslow (1996), the competency to write and read a program is less related with each other than it is expected to be. Writing and reading a program should be considered separately. Therefore, writing and reading the codes are investigated separately in the 4-stage method.

There is a model proposed by Xuewen & Ling (2011) to improve the programming skills including analysing and designing capabilities of students. According to the authors, good quality teaching materials have a certain effect on learning rates. This concept is also advised in the 4-stage method to improve programming skills of students. The teachers should cover the leading materials during the teaching of some important techniques such as dynamic programming, backtracking method, divide and conquer and brute force techniques. It is also suggested that this course should be supported with qualitative practical examples. The students should be given the ability to make decisions between the algorithmic techniques. The examples must be selected in a way to compare the strengths and the weaknesses of the different

methods.

One of the proposed methods in teaching ADA is using black box (Capay & Magdin, 2013). The students do observations by supplying various inputs and comparing the corresponding outputs. It is expected that students understand the concept of the algorithm which is specified as a black box. This method is quite time consuming, due to multiple and repeated entries of the students. However, the method itself stimulates the students to have logical thinking.

Instead of implementing classical teaching approaches, one of the proposed methods uses a computer environment (Begosso et al., 2012). It is aimed to bring the students' attention to a program by providing an interactive platform and showing the output immediately. A supporting program contains some animations and the students' interactions are necessary during the problem solving. According to the authors, providing such an interactive program to the students increases the attention of the problems rather than dealing with non-attractive problems.

The rest of the paper is organized as follows: Section 2 gives the details of methodology used and also explains each step of the 4-stage method. Section 3 shows the data collection process, results before and after the application of the 4-stage method including the discussion of the results. Section 4 gives the conclusions of the paper. Lastly, Section 5 outlines the recommendations.

2. METHODOLOGY

The students are required to be supervised by their instructors in the earlier levels of learning. However, as constructivists recommend, support is slowly withdrawn as the students master the topics. In contrast to a common belief, constructivism implies that the students feel more confident in programming based courses when they build up from the basics instead of having all necessary materials from their instructors (Ben-Ari, 1998).

Our teaching methodology is built up by applying 4 consecutive stages. Some of the stages as individual stages are still in use in some universities in teaching ADA. However, a blended version of them; applying 4-stages consecutively at a time is only practiced in our method according to our knowledge. The stages can be summarized as; understanding the concepts, formation of an algorithm using any programming languages, supplying various sets of problems and shuffling the sets of problems.

The method that was applied to validate the success of the teaching method can be outlined as follows. First, data was obtained from a cohort student group through surveys before the 4-stage method is applied. After the application of the method, surveys were conducted again to the similar cohort group of students and the comparison was performed between the two groups. The data was collected anonymously. To ensure reliability in data, a similar group of students (senior computer and software engineering students) were chosen in the years before and after the application of the 4-stage method. The students' success rates in two years were also analyzed and compared with their course learning outcomes survey results to ensure the validity of data.

2.1. The proposed 4-stage method

2.1.1. Understanding the concepts

It is necessary to give a set of problems related to the topics. However, even in the beginning of the course, the students deal with the difficulties to understand the concept of analysing and designing of algorithms. Basic concepts should be explained without using any programming language. This method is used to reassure the confidence of the students before any implementation. The steps used in this stage are shown in Figure 1 in the form of a flowchart.

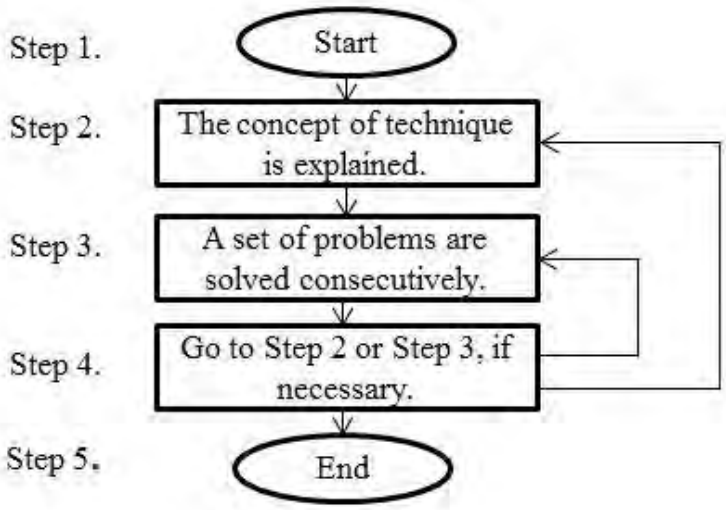

Figure 1. Understanding the concepts.

2.1.2. Formation of an algorithm

This second stage is implemented upon the completion of the first stage and right after the tutorial class for deeper understanding of analysing and designing techniques. To the instructor's preference, this method can be used together with the tutorial class. However, we suggest that the instructors should apply this stage after an efficient tutorial class. This methodology, shown in Figure 2, encourages the students to select an efficient algorithm design technique according to a given problem.

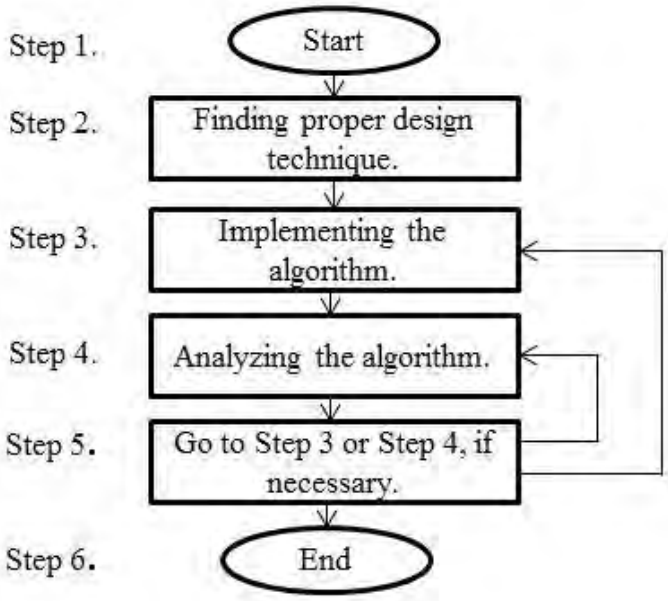

Figure 2. Formation of an algorithm.

2.1.3. Supplying various sets of problems

Since every student's learning potentials differ, a different set of problems are recommended to be given to each student. In a tutorial class, the examples start from the basic level and as the students comprehend the concept, more advanced examples are given. However, some students may not feel competent yet after analysing a few basic or intermediate examples. Accordingly, their concentration level can diminish easily during the tutorial class. To keep the students' attention alive, different sets of examples with different difficulty levels are distributed in the laboratory classes. The students who solve the problems by themselves are more willing to learn. Here, the most challenging task is to analyse the levels of each student to observe their weekly improvement and to assign the tasks. If any of the students feel inadequate even after having a specific level of task, the instructor provides another set of the same level of example. However, this request should be made by the students. Thus, the option to move on to the next level is given to the students and by this self-evaluation of each student is aimed to be developed. This stage can be seen in Figure 3 in the form of a flowchart.

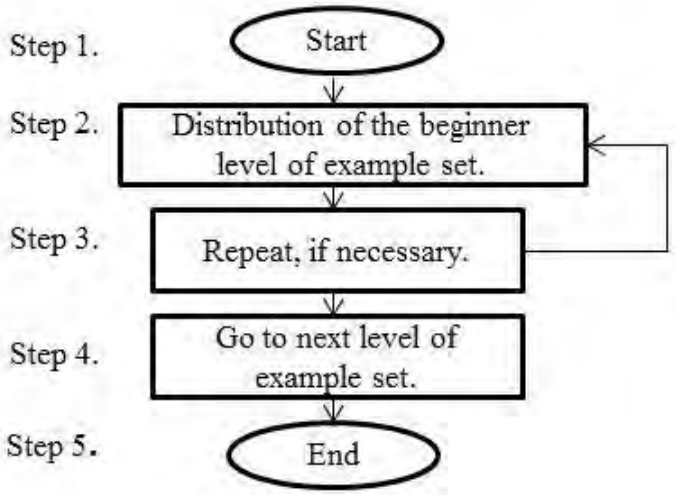

Figure 3. Supplying various set of problems.

2.1.4. Shuffling the set of problems

As it is assumed in the traditional teaching methods, all of the students in algorithm share the same levels. However, in the last step, the problem sets are distributed to the students as shuffled to break down this perception. In a way, at this level diagnostic teaching methodology and assessment method is applied. After applying stage 3, a new set of assignments are distributed to the students as shuffled. The distributed set contains different kinds of problems in 3 levels; beginner, competence and proficiency. Therefore, each student may have a chance to solve the problems at different levels at a time. In this stage, there is limited time duration for solving each level of problem. If no solution is received from the students within the limited duration, a new problem set is given without collecting the solution of the previous one. Therefore, stage 4 is only applicable, when the students have gained enough competency in solving all levels of problems. The observation of students' levels must be done by the instructor carefully with the intention to build the self-confidence of students. However, if students proceed to the next level as a result of wrong analysis done by the instructor, instead of improving their self-confidence it creates the loss of interest in programming. Because of such problems that can be faced during the implementation of stage 4, the instructor needs to give extra attention to a prior stage; stage 3. Figure 4 shows the principle of this stage in detail.

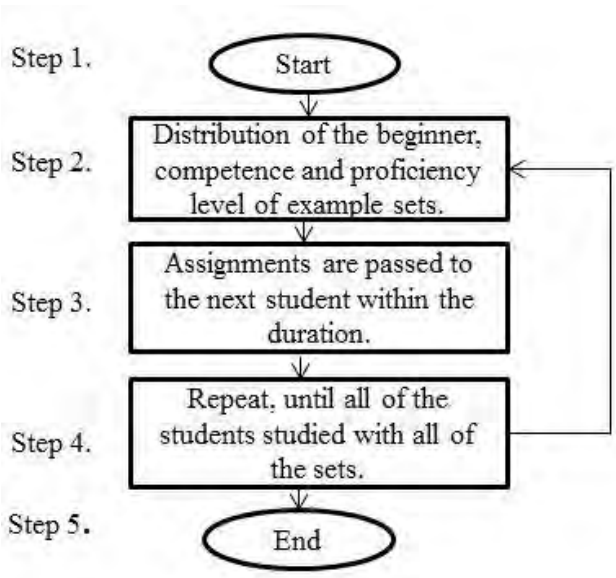

Figure 4. Shuffling the set of problems

3. RESULTS AND DISCUSSIONS

In the evaluation process, both summative and formative assessments were conducted. Firstly, as a formative assessment, the students evaluated themselves in terms of whether they reached the learning outcomes of the course. Then secondly, as summative assessment, the instructors collected the achievement rate of each student according to their grades obtained from their written exams at the end of the semester.

Learning outcomes of ADA were prepared in line with industry needs and adapted to the form of the learning outcomes of some well-ranked universities.

- LO1. Analyse the running time of iterative algorithms.
- LO2. Analyse the running time of recursive algorithms.
- LO3. Apply algorithm design methods including recursive, divide and conquer, dynamic programming and backtracking.
- LO4. Calculate the algorithmic complexity of the algorithms; worst-case and average-case behaviour.
- LO5. Differentiate the performance according to the choice of the algorithm design method.

At the end of the semester, students' evaluations and the instructor's remarks were collected to decide the success rate of our 4-stage method. The following remarks were noticed for the use of our method in teaching ADA course.

- Describing the preliminary steps such as data structure to be used or input size to be set for the problem at hand increased the success rate of students.
- Assisting to find analysing and designing techniques at the initial stages made the students understand the topics better.
- Assisting the students and gathering their responses required more effort and time for the instructor. However, this approach helped the students to progress and to feel more confident in analysing

and designing the algorithms.

In this case a survey was conducted among the students taking the course to evaluate their own opinion on achieving the learning outcomes. In the surveys, there were five responses; strongly agree, agree, uncertain, disagree and strongly disagree. In order to represent the outcomes accurately, strongly agree/agree answers were used together as satisfied, disagree/strongly disagree answers were represented as not satisfied, and uncertain answers were treated as neutral. When the feedback was compared with the previous semester's feedback, it can be seen that with the application of the new 4-stage method the achievement rate of the learning outcomes substantially increased and at least 84% of the students thought that they achieved the learning outcomes of the course. Figure 5 shows the achievement percentage for all of the learning outcomes before and after applying the 4-stage method.

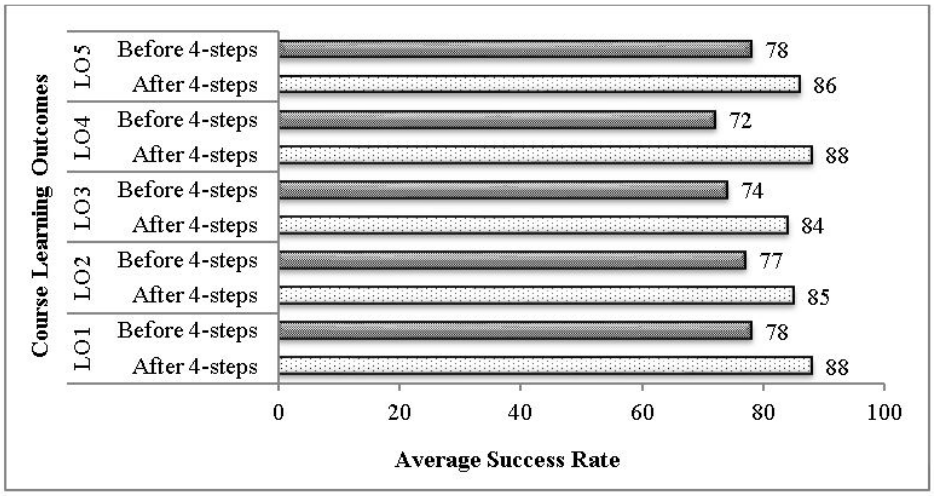

Figure 5. Learning outcomes feedback analysis before and after applying 4-stage method.

The instructor evaluated the achievement rate by referencing the students' grades collected for the semester. Accordingly, 62% of the class got grade C or higher while 21% of students passed the course with an average and 17% got F or FA. In the previous semester, where the same course was given without using the 4-stage method, the success rate was 45%. Meanwhile, 26% of the students passed with an average and 29% got F or FA. It is important to mention that the level of questions asked were about the same level in both years.

The students were also asked to evaluate their instructors and course in a questionnaire in terms of instructor capabilities and how he/she handles the course. In the survey, there were 10 related questions and the responses are given in Table 1. Among the 10 questions, the first seven questions are related to instructor's attributes. For example, is the instructor well prepared for each class or are the learning methods used effectively. Here, the lowest satisfaction rate of students was 78.57%, the highest dissatisfaction rate was 7.14% and it is seen that there is a noticeable amount of neutral rate among the answers. The rest of the questions were about students' opinions such as do they suggest the course to the other students or do they consider to take other courses from the same instructor. The lowest satisfaction rate was 81.48%, the highest dissatisfaction rate was 3.70% and it is noticed for the first seven questions, that there was a considerable number of students who responded as neutral. According to the students' responses, the following observations can be done.

- The satisfaction rate varied from 78.57% to 85.71% which shows that the 4-stage method was

accepted by most of the students and they were generally satisfied regarding the way of handling the ADA course.

- There was a considerable number of students who gave neutral response in the survey form which varied from 10.71% to 17.86%. It is known that the course was found difficult by the students. As a result of the failure, drop-out rates were high in the previous semester. The instructor encouraged the students to participate during the lecture, but still there were some students who had difficulty in getting motivated for the class. This is probably the reason of the neutral responses in the survey.

- The rate of dissatisfaction changed from 0-7 to 14%. Even after applying the 4-stage method for one semester, it was still not an easy task to design and to analyse the algorithms for some of the students. Therefore, it requires a couple of semesters for the students as well as the instructor to adapt this new technique.

- Despite having a high satisfaction rate to the question about how the instructor used instructional materials, it had the lowest satisfaction rate in the survey. It is worth mentioning that, in all of the stages, active learning methods were employed by the instructor such as discussions, problem solving activities etc. However, it is known that 4-stage method requires extensive instructor effort especially in collecting the assignments and evaluating the students' levels regarding their performances. The attention was given to those stages rather than the use of various types of instructional materials and it may be the possible reason of the low satisfaction rate for the related question.

Table 1. Distribution of the students' answers in the satisfaction survey.

Questions	Satisfied	Neutral	Not Satisfied
The instructor seemed well prepared for each class and laboratory.	81.48%	14.81%	3.70%
The instructor attended his/her classes/laboratories regularly.	85.18%	14.81%	0
The instructor encouraged the student participation by using appropriate methods.	85.71%	10.71%	3.57%
The instructor successfully employed active learning methods. (e.g. discussions, problem-solving activities and experiment conducting.)	81.48%	14.81%	3.70%
The instructor used instructional materials efficiently.	78.57%	14.29%	7.14%
The instructor followed the course plan during teaching and lab sessions.	85.71%	14.29%	0
The instructor was fluent and effective in teaching.	82.14%	14.29%	3.57%
I would suggest this course to other students.	85.18%	11.11%	3.70%
This course was useful for my professional and personal development.	82.14%	17.86%	0
I would consider taking another course from the instructor.	81.48%	14.81%	3.70%

4. CONCLUSIONS

In this work, the 4-stage method is introduced to enhance the students' problem solving ability and to increase their programming skills for the programming courses; for instance, Analysis and Design of the Algorithms (ADA) course.

After the application of the 4-stage method for one semester, it is observed that problem solving ability has increased in comparison to the previous semester. Even in the beginning of the semester, the students had lack of competency in writing the basic concepts of programs. After the application of the 4-stage method, 85.18% of the students recommended this course to the other students at the end of the semester.

Distribution of the multiple assignments with different difficulty levels and evaluating the students' performances after each assignment made students more competitive, eager to learn and interested. This is beside the fact that some students were neutral or disagreed regarding the implementation of the method, 78.57% of them were satisfied with the course. Employment of the 4-stage method also helps the instructor's proficiency in analysis and design of the algorithms. Preparing different levels of questions for each laboratory, evaluating questions and assisting the students enhances teaching ability of instructors. On the other hand, it is important to mention that this method gives extra work load to the instructors especially during assignment distribution and evaluating.

Application of each stage requires more instructor concentration than application of traditional techniques. As a result of this, the instructors were not able to use different instructional materials as it is seen in the students' feedback. In order to ease the burdens of the instructors, it is required to perform stage 3 and stage 4 together with the teaching assistants.

Although the survey results indicated remarkable satisfaction in general, it was noticed that there were some neutral and non-satisfactory responses as well. The 4-stage method implementation is suggested for the upcoming semesters to reduce the non-satisfactory responses further.

5. RECOMMENDATIONS

Further recommendations to this research can be provided as three main points. As a first point the proposed 4-stage method needs to be applied for two more years to see the success rates and survey results of students. As a second point a survey can be conducted among the instructors who are teaching the 4-stage method. This may give a flexibility to the instructor to develop the 4-stage method. Another recommendation point can be exploring different ways to measure the course learning outcomes and comparing these values with the survey values to ensure stronger validity on results.

Conflict of interests

The authors declare no conflict of interest.

REFERENCES

Begosso, L. C., Begosso, L. R., Goncalves, E. M., & Goncalves, J. R. (2012). An approach for teaching algorithms and computer programming using Greenfoot and Python. *In 2012 Frontiers in Education Conference Proceedings (pp. 1-6). IEEE.* https://doi.org/10.1109/FIE.2012.6462307

Ben-Ari, M. (2001). Constructivism in computer science education. *Journal of Computers in Mathematics and Science Teaching, 20(1), 45-73.*

Bonar, J., & Soloway, E. (1985). Preprogramming knowledge: A major source of misconceptions in novice programmers, *In E. Soloway & J.C. Spohrer (Eds.), Studying the novice programmer (pp. 261-279). Hillsdale, NJ: Lawrence Erlbaum.*

Capay, M., & Magdin, M. (2013). Alternative methods of teaching algorithms. *Procedia-Social and Behavioral Sciences, 83, 431-436.* https://doi.org/10.1016/j.sbspro.2013.06.085

Dreyfus, H. L., & Dreyfus, S. E. (1986). The power of human intuition and expertise in the era of the computer. *Mind over machine. Nueva York: The Free Press.*

Du Boulay, B. (1986). Some difficulties of learning to program. *In E. Soloway & J.C. Spohrer (Eds.), (pp.283-299). Hillsdale, NJ: Lawrence Erlbaum.*

Hamzat, A., Bello, G., & Abimbola, I. (2017). Effects of computer animation instructional package on students' achievement in practical biology. Cypriot Journal of Educational Sciences, 12(4), 218-227. https://doi.org/10.18844/cjes.v12i4.2932

Hooshyar, D., Ahmad, R. B., Yousefi, M., Fathi, M., Horng, S. J., & Lim, H. (2018). SITS: a solution-based intelligent tutoring system for students' acquisition of problem-solving skills in computer programming. *Innovations in Education and Teaching International, 55(3), 325-335.* https://doi.org/10.1080/14703297.2016.1189346

Levitin, A. (2008). Introduction to Design and Analysis of Algorithms, *2/E. Pearson Education India.*

Lin, H. C. (2006). An Internet-based graphical programming tool for teaching power system harmonic measurement. *IEEE Transactions on Education, 49(3), 404-414.* https://doi.org/10.1109/TE.2006.879239

Milne, I., & Rowe, G. (2002). Difficulties in learning and teaching programming—views of students and tutors. *Education and Information technologies, 7(1), 55-66.* https://doi.org/10.1023/A:1015362608943

Nogueira, J. R., Alves, R., & Marques, P. C. (2019). Computational Programming as a Tool in the Teaching of Electromagnetism in Engineering Courses. *Improving the Notion of Field. Education Sciences, 9(1), 64.*

Perkins, D. N., Hancock, C., Hobbs, R., Martin, F., & Simmons, R. (1986). Conditions of learning in novice programmers. *In E. Soloway & J.C. Spohrer (Eds.), Studying the novice programmer (pp. 261-279). Hillsdale, NJ: Lawrence Erlbaum.*

Winslow, L. E. (1996). Programming pedagogy—a psychological overview. *ACM Sigcse Bulletin, 28(3), 17-22.* https://doi.org/10.1145/234867.234872

Xuewen, X., & Ling, G. (2011). Notice of Retraction Teaching method innovation in" algorithm analysis and design". *In 2011 International Conference on E-Business and E-Government (ICEE) (pp. 1-3). IEEE.* https://doi.org/10.1109/ICEBEG.2011.5881621

Yıldız, E., Alkan, A., & Cengel, M. (2019). Teacher candidates attitudes towards the stem and sub-dimensions of stem. Cypriot Journal of Educational Sciences, 14(2), 322-344. https://doi.org/10.18844/cjes.v14i2.4144