Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Towards enhancing teaching and learning computer programming in Saudi Arabia

Ahmed Abba Haruna *

Department of Computer Science, College of Computer Science and Engineering, University of Hafr Al Batin, Hafr Al Batin, Saudi Arabia

ARTICLE INFO

Article history:
Received 7 March 2022
Received in revised form
6 December 2022
Accepted 29 December 2022

Keywords:
Improving computer programming teaching
Learning
Student-centered
Teaching pedagogy
Flip classroom

ABSTRACT

Globally, numerous students struggle with understanding computer programming, which is one of the most difficult courses in the computer science curriculum as stated in the literature. This is especially true in a diverse learning environment where students come from different disciplinary backgrounds, language skills, and cultures. Hence, to improve on the aforementioned challenges, another research introduces a framework that combines constructivist and collaborative learning theories with a student-centered teaching pedagogy for teaching postgraduate introductory programming classes at Central Queensland University, Australia. However, the framework will not work effectively when teaching computer programming courses to undergraduate students in Saudi Arabia. This is due to geographical differences, study levels, negative emotional issues, and stress affecting students' learning. Such as students' first view that programming is difficult, or the difference between students' and teachers' perspectives on learning (disciplinary backgrounds, language skills, and cultures). Therefore, this paper proposed a student-centered learning and teaching method that combines constructive alignment (consistency), collaborative learning theory (collaboration, conception, and cognition), and bits of Maslow's theory (love/belonging and self-actualization) in a student-centered teaching pedagogy. The research findings reveal that when using the existing method to teach Introduction to Computing-II (Object Oriented Programming in Java) at the University of Hafr Al Batin, Saudi Arabia, only 87.5% of students passed the course, while 12.5% failed. However, when using the proposed method 95.2% of the students passed the course, while only 4.8% failed. Thus, the proposed method clearly shows significant improvement, with the failure rate reducing from 12.5% to only 4.8%.

1. Introduction

Computer programs operate all computer-based technology that we use today. It also defined the functionality of most equipment, such as personal computers, a television, an airplane, and mobile phones, among others. Computer programs create and explain the functions of the devices in which they are embedded, as well as provide us with a way to operate them. One of the definitions of a structured programming paradigm is that it depicts a computer program as a set of tasks. Any activity that is too complicated to be described is broken down into a series of smaller component activities until each task is small and self-contained enough to be easily understood (Konecki and Kadoić, 2015). According to Kanaparan et al. (2019), Iskrenovic-Momcilovic (2018), and Figueiredo and García-Peñalvo (2018), many students find difficulty in learning programming language, and many among them struggle to understand programming concepts. Students find programming challenging because they don't comprehend fundamental programming principles and techniques. Early failure to grasp the crucial concepts of programming undermines students' confidence and raises the dropout rate (Krpan et al., 2015). Numerous computer programming teaching methods, such as the ones described below, have been developed and used as a result of the challenges in teaching and learning computer programming.

* Corresponding Author.
Email Address: aaharuna@uhb.edu.sa
https://doi.org/10.21833/ijaas.2023.03.024
Corresponding author's ORCID profile:
https://orcid.org/0000-0003-2488-7438

1.1. Motivation and learning styles

University academic affairs state that the efficiency of any learning strategy is dependent on the student's learning styles and motivation. As a result, those factors have an impact on programming learning. "In the context of learning, a desired learning mode is one in which students respond to and use stimuli" (Kanaparan et al., 2019). "Some students are visual learners, while others are aural or kinesthetic learners" (Mohorovicic and Strcic, 2011). A learner's ability to develop programming abilities fast and readily may be aided by selecting a specific learning style or type of motivation. In contrast, "if a learner uses the incorrect technique or lacks motivation, learning to program may be tough" (Jenkins, 2002). Consequently, Jenkins (2002) classified learning styles into two groups: Deep approach and surface approach. Surface learning is defined as memorizing data, while deep learning is defined as gaining comprehension of a topic. Surface learning is useful for memorizing a programming language's syntax, but deep learning, in addition to surface learning, is necessary for gaining a true understanding of programming logic and, as a result, true programming skills.

Moreover, extrinsic, intrinsic, and social motivation are all essential factors that influence learning efficiency. "Extrinsic motivation is obtained from expected external benefits, such as financial gains; intrinsic motivation is derived from within; and social motivation is formed from a desire to please a third party (family, teacher, friends, and so on)" (Jenkins, 2001; 2002). The level of motivation is influenced by two factors: Expectation and value (Jenkins, 2001). A student's value for success, such as future employment, is explained by a value, and anticipation is the student's own expectation of achievement. If a student does not value achievement or does not expect to pass the programming course, he may be unmotivated to complete course assignments. The teacher's function is not merely to transmit knowledge; he also serves as a motivator, ensuring that students engage in the assignments he has devised (Wang, 2022).

1.2. Methodology for teaching programming

Teachers utilize teaching as a form of instruction to provide course material. It is a technique for transforming information into learning that emphasizes the "how to" of training delivery" (Kanaparan et al., 2019). Teaching information and communication technologies (ICT), and hence teaching computer programming, is a novel problem domain compared to teaching more established courses like mathematics or physics. As a result, its teaching style is less well-established, resulting in most teachers using a mix of methods, with one method predominating (Papp-Varga et al., 2008).

Computer programming training methodologies are classified as statement-oriented, tool-oriented, software technology-related, task-type-focused, language-oriented, action-oriented, and sample task-based by one categorization (Papp-Varga et al., 2008). Different approaches cover the selected programming language to differing degrees and teach its features in a different order depending on the goals. Statement-oriented techniques, for example, regard the programming language to be a collection of statements, with each piece being taught in a certain order.

Language pieces are presented in the order in which they are necessary to handle genuine problems in task-type-oriented approaches. Techniques that present a programming language just for the amount required, taking into account a different principal aim, such as database instruction, are known as "using as a tool" methods. Programming can be taught in two ways: Bottom-up and top-down (Saito and Yamaura, 2013). The bottom-up method prioritizes teaching syntax and specific programming language features. More intricate constructions are considered after specific elements have been taught. Understanding abstractions, regardless of how they are physically implemented, is the first step in the top-down method. Students are taught implementations once they have grasped the abstractions and their utility.

2. Problem statements

In a diverse learning environment where students come from a variety of disciplinary backgrounds, language skills, and cultures, the present convolutions of teaching and learning computer science (computing) are heightened. The gap between students' and teachers' opinions on learning outcomes and task completion (Xu et al., 2021); students' initial view that "learning programming is tough" are some of the elements impacting computer science students' learning outcomes (Tan et al., 2009). Then there are the "components of environmental change" (disciplinary backgrounds, language skills, and cultures). Computer programming demands the application of advanced cognitive abilities such as logic, problem-solving, and planning (Lu and Zheng, 2022). A programmer abstracts a process, defines it as logic structures, and then translates it into precise code using a formal language (Krpan et al., 2015).

However, there are a number of learning theories or approaches that could be effective in methodically guiding computer science students so that they can learn and feel comfortable with new ideas and their application in a range of learning environments. Among the strategies used are constructive alignment, collaborative learning theory, and Maslow's (1974) hierarchy of needs theory. When a learner develops his or her own learning through relevant learning activities and the teacher creates a learning environment that supports learning activities that are relevant to accomplishing the desired learning objectives, this is referred to as constructive alignment (Biggs and Tang, 2011). Collaborative learning facilitates cognitive

constructivism by improving conceptual comprehension through peer debate (Mazur, 2017). The social production of new knowledge begins with a reflection on new material. Cognitive rehearsal and assimilation are made possible by teamwork. Deeper understanding, aided by collaboration among team members with varying levels of prior expertise (Sangin et al., 2011). Maslow's (1974) hypothesis is a psychological motivational theory that consists of a five-tiered hierarchy of human needs, as seen in Fig. 1.

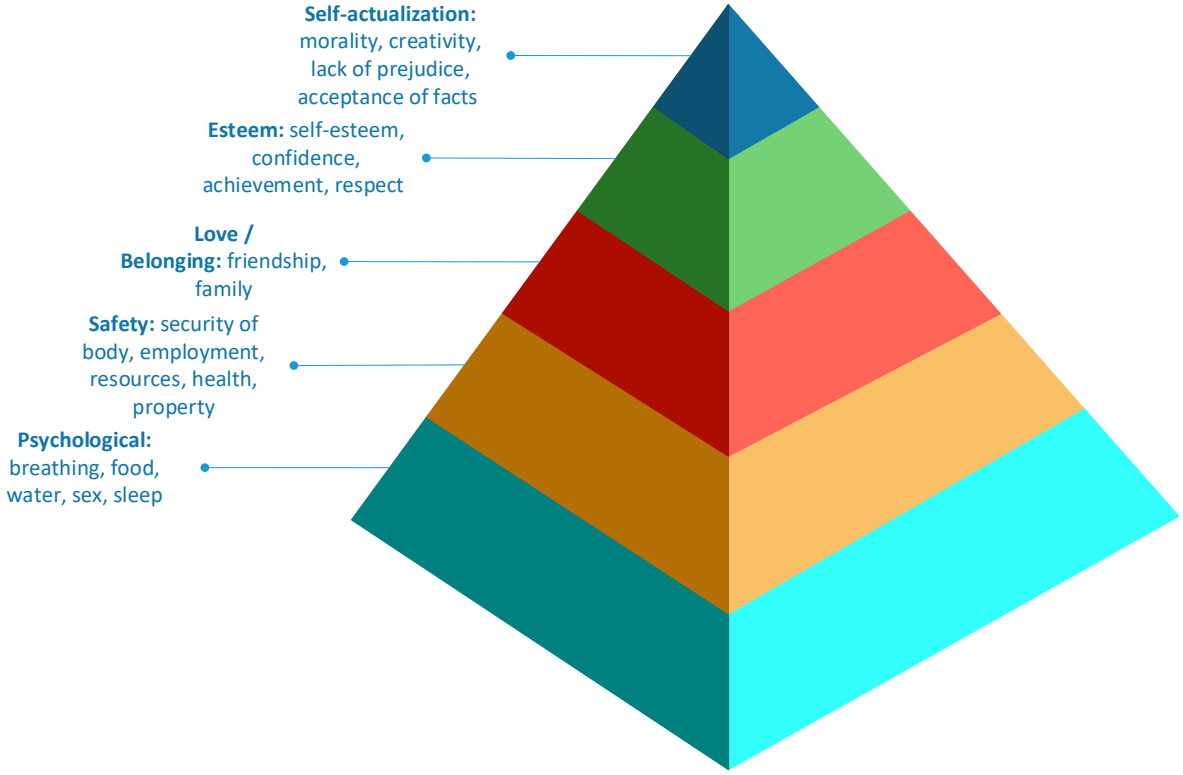

Fig. 1: Maslow's (1974) hierarchy of needs

A person must meet the five hierarchy of needs in the following order to be self-actualized: Physiological, safety, love and belonging, esteem, and self-actualization. By evaluating relevant material, this study must refute Maslow's (1974) theory of Humanism, arguing that most people (for example, a person from a poor family who rose through the ranks of society) attain self-actualization without traveling through the hierarchy of needs step by step. As a result, Maslow's (1974) hierarchy of needs hypothesis does not always apply; it varies because some people want self-actualization before needing health, safety, or belonging.

As a result, according to McLeod (2007), not everyone will advance through the hierarchy in a single direction, but will instead switch back and forth between the various forms of requests. Student ratings and grades improved after employing constructive alignment, according to Larkin and Richardson (2013) and according to Imenda (2018), there are instances when behavioristic instructional techniques, cognitivist strategies, constructivist strategies, and combinations of the three work well.

While traditional teacher-centric pedagogy focuses on course content and transferring knowledge to students, learner-centric pedagogy focuses on assisting students in developing or building knowledge (Wright, 2011). However, it is not thought to be effective in systematically guiding computer science students to be able to learn and feel very comfortable with new ideas and their applications. As a result, moving from a teacher-centered to a learner-centered design is recommended (Smart et al., 2012). As a result, students must develop soft skills such as collaboration, communication, critical thinking, and creativity, the latter of which is one of the most in demand by employers (Welkener, 2013).

Hence, to improve on the aforementioned challenges, Tom (2015) introduced a five C framework, which combines constructivist and collaborative learning theories with a student-centered teaching pedagogy for teaching postgraduate introductory programming classes at Central Queensland University, Australia. However, the framework (Tom, 2015) will not be applied effectively when teaching computer programming courses to students in Saudi Arabia. This is due to geographical differences, study levels, negative emotional issues, and stress affecting students' learning. Such as students' first view that programming is difficult, or the difference between

students' and teachers' perspectives on learning (disciplinary backgrounds, language skills, and cultures).

While social, intrinsic, and extrinsic motivation are all critical elements that affect learning effectiveness (Jenkins, 2001; 2002). Extrinsic motivation stems from anticipated external rewards, such as money gains, whereas intrinsic motivation comes from within, and social motivation is fueled by the need to appease others (such as family, teachers, friends, and others). Hence, it is necessary to incorporate Maslow's (1974) theory (love/belonging and self-actualization) into a student-centered teaching pedagogy, because a teacher's role extends beyond merely disseminating information. Also, it's essential for a teacher to serve as a motivator, ensuring that students complete the practices, exercises, and assignments he has created (McLeod, 2007).

Therefore, this research proposes a student-centered learning and teaching technique based on constructive alignment (consistency), collaborative learning theory (collaboration, conception, and cognition), and Maslow's (1974) theory (love/belonging and self-actualization) in a student-centered teaching pedagogy. This research aims to apply the proposed framework to bachelor's degree students at a Saudi university.

3. Proposed methodology

Based on the concerns raised in Section 2, this paper proposes a student-centered learning and teaching method that incorporates constructive alignment (consistency), collaborative learning theory (collaboration, conception, and cognition), and elements of Maslow's (1974) theory (love/belonging and self-actualization) in a student-centered teaching pedagogy, as shown in Fig. 2.

Fig. 2 is designed to turn a traditional two-hour lecture into a flipped classroom (by assigning recorded video lectures as homework and using class time for active learning exercises/activities and direct engagement/discussion with students), however, it is broken into five sessions:

1. Explanation/Elaboration
2. Conceptualization and communication
3. Interaction
4. Collaborative problem solving
5. Presentation and demonstration

That is suited to the demands of student cohorts with a variety of disciplinary backgrounds and learner profiles.

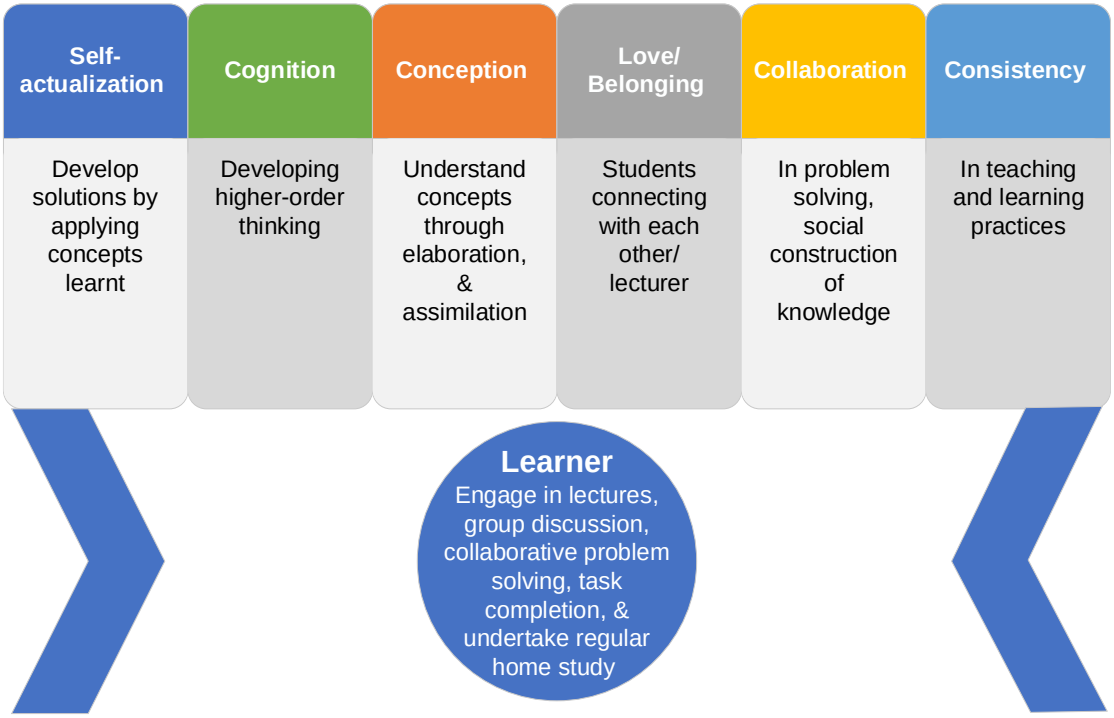

Fig. 2: Integration of learning theories into student-centered teaching pedagogy

The sessions in each pack below are based on one or more of the six processes shown in Fig. 2:

1. Explanation/Elaboration Session: By explaining when appropriate, elaborating on the subject to be taught using simulations and real-world examples, and referring to previously learned concepts ("teaching by facilitating") (Rogers et al., 1967), which is "driven by what learners need to know" (Knowles et al., 2005).
2. Conceptualization and Communication Session: By devoting 5 to 10 minutes to small group conversations, this program encourages collaborative learning. Questions regarding the main notion will guide group conversations, allowing for the social generation of new knowledge "learners employing prior experience"

(Knowles et al., 2005). These communication sessions will help students frame inquiries and articulate their uncertainty more easily in the early stages of the course since "learning is ubiquitous-integrating the whole person" (Rogers et al., 1967).

3. Interaction Session: Students will ask questions or share their discoveries from group discussions during the interaction session ("which is developing learner's full potential") (Rogers et al., 1967). Based on this input, the new concept is further articulated and related to existing schemas in the knowledge development process ("by determining what has to be learned, which is relevant") (Knowles et al., 2005). Students will also be able to tell the rest of the class about their group learning experiences.
4. Collective Problem-Solving Session: Students will ask questions or share discoveries from group discussions during the interaction session ("which is developing learner's full potential") (Rogers et al., 1967). Based on this information, the new concept is further articulated and related to current knowledge production schemas ("by determining what has to be learned, which is relevant") (Knowles et al., 2005). The remainder of the class will be able to hear about the students' group learning experiences.
5. Presentation Session: During this session, students will get the opportunity to showcase their own developed programming application to the class and discuss the source code. Students will be encouraged to be more creative during this session. It will also inspire students to learn outside of the classroom ("increasing personal engagement through internal motivation") (Rogers et al., 1967), which is linked to "self-esteem" (Knowles et al., 2005), and "creating unconditional positive respect" (Rogers et al., 1967).

4. Results and discussion

A multipartite strategy will be used to investigate the outcomes and impact of the proposed methods.

1. Survey: This is done by conducting a survey to confirm the emotional concerns that students face while studying programming. In addition, using the six phases outlined in Fig. 1, collect feedback on the efficacy and student experiences. Specifically, to assess the impact of the aforementioned sessions, which were created using the six-step process.
2. Grades: This is done by comparing student performance across class sections over the course of the semester with and without the suggested technique.

These evaluation procedures would reveal/validate the suggested method's potential for improvement.

4.1. Survey

The survey was done for the course code CSE201, also known as Introduction to Computing-II (Object Oriented Programming), at the University of Hafr Al Batin, Saudi Arabia during the academic year 2021-2022, Term 1. A total of 38 students participated in the evaluation.

This study used ten (10) different performance metrics to evaluate the proposed student-centered learning and teaching method that was integrated into the five sessions stated in section 3 (2 metrics per session as shown from Fig. 3 to Fig. 7). In order to do the analysis, five keywords were employed per metric: Strongly agree; agree; neutral; disagree; strongly disagree.

Hence, Fig. 3 combines constructive alignment (consistency) with conception and cognitive aspects. Consequently, 81.6% of students strongly agreed that the instructor was good at explaining concepts using real-life situations, whereas 18.4% agreed. Only 60.5% strongly agreed that the course content was adequately organized and prepared, while the remaining 28.9% agreed. However, 10.5% of students remained unsure. As a result, the findings indicate that the session's goal was met in a significant degree.

Fig. 3: Evaluation based on explanation/elaboration session

Fig. 4 demonstrates that 81.1% of students strongly agreed that the instructor provided opportunities for students to collaborate with one another as part of this module, while 5.4% only agreed. 13.5% of students, on the other hand, were undecided. In comparison, only 68.4% strongly felt that the course load was acceptable, with the remaining 15.8% agreeing but not strongly. While

10.5% of students were undecided, 5.3% were vehemently opposed.

Students improve their learning skills by participating in group conversations, assessing information, and applying concepts to solve problems. When students are given the opportunity to construct, develop, and present their own applications to the class, they are motivated to work more and become more interested in learning. It may also aid students in developing their lifetime learning and research skills.

It is clear that the integration of constructive alignment (consistency) and collaborative learning theory (collaboration, conceptualization, and cognition) has significantly improved students' learning experiences.

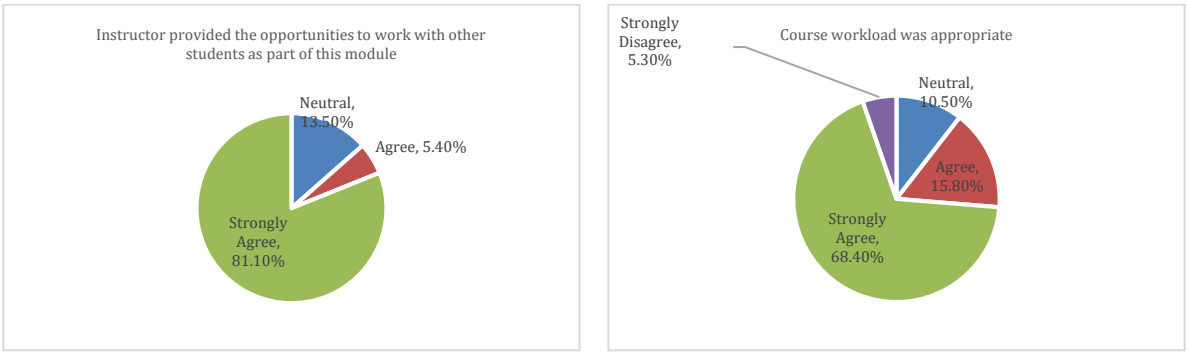

Fig. 4: Evaluation based on conceive and communication session

Fig. 5 depicts the combination of constructive alignment (consistency) with Maslow's (1974) hierarchy of needs (students experiencing love/belonging and self-actualization). As a result, 86.8% of students strongly agreed that the instructor values students' ideas and opinions, while 13.2% just agreed. Only 73.7% strongly agreed that the course was designed to allow all students to fully participate, with the remaining 23.7% agreeing but not strongly. In contrast, 2.6% of students were undecided. As a result, the findings show that the session's goal was met to a large extent.

Fig. 5: Evaluation based on interaction session

Fig. 6 also displays the intersection between collaborative learning theory (collaboration, conceptualization, and cognition) with Maslow's (1974) hierarchy of needs (students experiencing love/belonging and self-actualization). According to Fig. 6, 89.2% of students strongly agreed that the instructor provided them with opportunities to apply what they had learned, whereas 10.8% agreed. Only 83.8% of respondents strongly agreed that the instructor was available and helpful, with the remaining 13.5% agreeing barely. 2.7% of students, on the other side, were undecided. As a result of the findings, the session's purpose was met to a considerable extent.

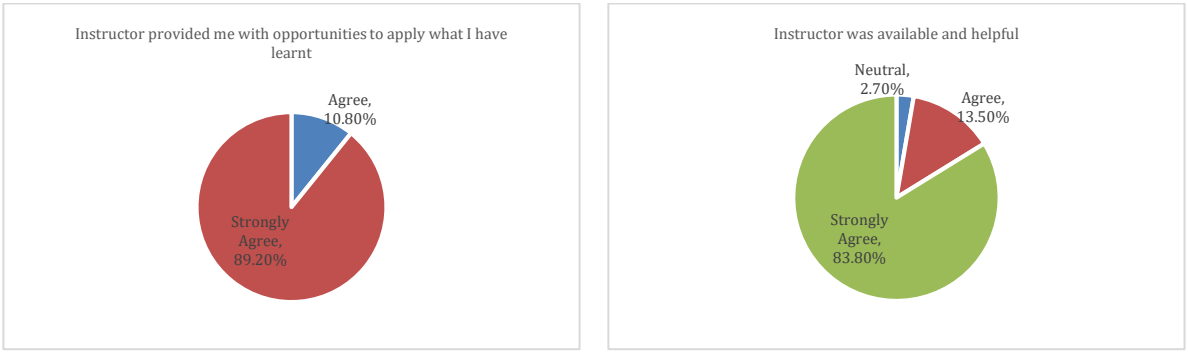

Fig. 6: Evaluation based on collective problem-solving session

Fig. 7 combined constructive alignment (consistency), collaborative learning theory (collaboration, conception, and cognition), and Maslow's (1974) hierarchy of needs (love/belonging and self-actualization). According to Fig. 7, 89.2% of students strongly agreed that the instructor piqued their attention, whereas 10.8% just agreed. Only 68.4% of respondents strongly agreed that the learning objectives were obvious, with the remaining 28.9% agreeing although not strongly. On the other hand, 2.6 percent of students were skeptical. As a result, the results indicate that the session's goal was significantly met.

Fig. 7: Evaluation based on a presentation session

The analysis of the data from the collected interview responses in Fig. 3 to Fig. 7 clearly shows how the integration of Maslow's (1974) theory concepts of love/belonging and self-actualization, as well as collaborative learning theory's concepts of collaboration, conception, and cognition, has benefited the learning process and enhanced the student learning experience. It's evident that students had the anticipated effect of easing or lessening the initial stress, worry, or fear associated with studying. Learning has also become more fun for students as a result of encouraging more passion.

4.2. Grades

A grade evaluation was also conducted to further investigate the effectiveness of the proposed method in comparison to the existing method (Tom, 2015). Table 1 illustrates a comparison of the academic performance (grades) of students when teaching Introduction to Computing-II (Object Oriented Programming) using the proposed method vs teaching without using the proposed method (Tom, 2015). Table 1 shows two cohort results that were comparable because they were composed of a similar mix of undergraduate students from diverse disciplines (Cyber Security, Software Engineering, and Computer Science).

The analysis shows that when utilizing the benchmark approach (Tom, 2015) in teaching CSE201, only 87.5% of students passed the course, while 12.5% failed. When the proposed method was utilized in teaching the same course, however, 95.2% of students passed and only 4.8% failed. Consequently, the proposed method clearly shows significant improvement, with the failure rate reducing from 12.5% to only 4.8%. This demonstrates that it is possible to increase student performance because the proposed method has assisted the students at University of Hafr Al Batin in dealing with many of the emotional issues that affect their performance and learning has become a more enjoyable experience, which in turn encouraged them to persevere in their efforts.

Furthermore, Table 2 also provides some students' feedback on the proposed strategy for teaching Introduction to Computing-II (Object Oriented Programming). This highlighted the effectiveness of integrating the bit of Maslow's (1974) theory (love/belonging and self-actualization) into the proposed method because a teacher's duty extends beyond simply conveying information. He must also act as a motivator, ensuring that students' learning is more enjoyable and successful. Thus, it is evident that the gap between the student's and the teacher's perspective of learning (disciplinary backgrounds, language skills, and cultures) is managed effectively.

While the proposed method has proven to be very effective and efficient when applied to a cohort of min 30 to 40 students, it may not be as effective and efficient when applied to a cohort of 300-to-3000 students because Class size is too large for effective facilitation; less effective for physical and real-time interactions between lecturer and students; challenging and less effective for assessments and feedbacks. Massive Open Online Course (MOOC) techniques, on the other hand, may be utilized to scale delivery methods and foster active and deep learning (Hamal, 2022).

Table 1: Data comparing the proposed method to the existing method in terms of the academic performance (grades) of students who took introduction to computing-II (Object-oriented programming)

Term/Year	Course name	Pass rate (%)	Failure rate (%)
Term 1 2021	Introduction to Computing-II -Using the proposed method	95.8%	4.8%
Term 1 2021	Introduction to Computing-II - Using existing method (Tom, 2015)	87.5%	12.5%

Table 2: Some of the students' feedback on the proposed strategy for teaching introduction to computing-II

No. students	What aspects of my teaching were most useful or valuable?
1	Very passionate and honest, Very respectful and humble towards his students. The relationship felt more like a father and son rather than a teacher and a student. I wish the best and hope to get education from him again
2	The way you taught your students was very kind and very good
3	The information during teaching is clear
4	That you explain the code line by line and give us example from real life so we connect them together and never forget it
5	Nothing in particular, but clear and organized
6	That you were careful and helpful
7	The teacher was extremely understanding of the students' position so he took good care of us and gave us more than enough motivation which is something not many teachers are willing to give. Overall the teacher is definitely one of the best teachers the university has to offer and words can't describe him enough
8	everything he said is useful and valuable to me as a student I wish him all the best and I will try my best to learn from him
9	The way you teach is so impressive, when you ask about the student, how they feel, what they need, help them, even learn from them, make them feel that they have value, laughing with them, this is what being a teacher/ lecturer is I have a lot of words but I can't write all of it here, hope you read this. Thank you very much.
10	Everything he said is useful and valuable to me as a student
11	Examples from real or life
12	Inspiration, Exciting students to code and try.
13	The way you connected real life with class subjects
14	Organizing the lecture time and introducing real-life examples in the explanation of the material, as well as giving the students complete freedom to discuss any idea related to the lesson and respond to asking questions, in addition to constantly responding to students and their inquiries even during times outside the lecture hours.
15	Talk to students and understand what they want and hear their opinions
16	Your teaching and your way of helping us and trying of understanding us and you are amazing doctor you always try to help your student in everything you can
17	Collaboration and giving examples
18	All aspects of teaching

4.3. Validity issues

No experimental controls have been applied because this is an in situ study of independently managed university classes. The implementation of the proposed method and the data-gathering techniques utilized reasonably establish the validity of this work as per the benchmark approach (Tom, 2015).

The main campus of the University of Hafr Al Batin is where this study is being undertaken. The university comprises multiple geographically dispersed campuses. The course coordinator creates the assessment items and the essential course materials, and a structured moderation procedure is used to guarantee that the same standards are applied on all campuses. The evaluation items won't be altered or compromised as a result of this investigation, therefore it can be guaranteed.

5. Conclusions

While social, intrinsic, and extrinsic motivation are all essential components of learning efficacy. Hence, it is necessary to incorporate bits of Maslow's (1974) theory (love/belonging and self-actualization) into a student-centered teaching pedagogy, because a teacher's role extends beyond merely disseminating information. Also, it is essential for a teacher to serve as a motivator, ensuring that students complete the practices, exercises, and assignments he has created. Therefore, this study proposes a student-centered learning and teaching technique in a student-centered teaching pedagogy. The research findings reveal that the proposed method clearly exhibits a significant improvement, with the failure rate reduced from 12.5% to 4.8%. This demonstrates that it is possible to increase student performance because the proposed method has assisted the students at University of Hafr Al Batin in dealing with many of the emotional issues that affect their performance and learning has become a more enjoyable experience, which in turn encouraged them to persevere in their efforts. Thus, it is evident that the gap between the student's and the teacher's perspective of learning (disciplinary backgrounds, language skills, and cultures) is managed effectively.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Biggs J and Tang C (2011). Teaching for quality learning at university. McGraw-Hill Education, New York, USA.

Figueiredo J and García-Peñalvo FJ (2018). Building skills in introductory programming. In the 6th International Conference on Technological Ecosystems for Enhancing Multiculturality, Salamanca, Spain: 46-50. https://doi.org/10.1145/3284179.3284190 **PMid:28736922**

Hamal O (2022). Intelligent system using deep learning for answering learner questions in a MOOC. International Journal of Emerging Technologies in Learning, 17(2): 32-42. https://doi.org/10.3991/ijet.v17i02.26605

Imenda SN (2018). On the unity of behaviourism, cognitivism, and constructivism in teaching and learning. International Journal of Educational Sciences, 20(1-3): 86-95. https://doi.org/10.31901/24566322.2018/20.1-3.11

Iskrenovic-Momcilovic O (2018). Learning a programming language. International Journal of Electrical Engineering Education, 55(4): 324-333. https://doi.org/10.1177/0020720918773975

Jenkins T (2001). Teaching programming–A journey from teacher to motivator. In The 2nd Annual Conference of the LSTN Center for Information and Computer Science, London, UK.

Jenkins T (2002). On the difficulty of learning to program. In the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences (LTSN-ICS), Loughborough, UK: 53-58.

Kanaparan G, Cullen R, and Mason D (2019). Effect of self-efficacy and emotional engagement on introductory programming students. Australasian Journal of Information Systems, 23. https://doi.org/10.3127/ajis.v23i0.1825

Knowles MS, Holton E, and Swanson R (2005). The adult learner: The definitive classic in adult education and human resource development. 6th Edition, Elsevier, Burlington, USA.

Konecki M and Kadoić N (2015). Intelligent assistant for helping students to learn programming. In the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics, IEEE, Opatija, Croatia: 924-928. https://doi.org/10.1109/MIPRO.2015.7160406

Krpan D, Mladenović S, and Rosić M (2015). Undergraduate programming courses, students' perception and success. Procedia-Social and Behavioral Sciences, 174: 3868-3872. https://doi.org/10.1016/j.sbspro.2015.01.1126

Larkin H and Richardson B (2013). Creating high challenge/high support academic environments through constructive alignment: Student outcomes. Teaching in Higher Education, 18(2): 192-204. https://doi.org/10.1080/13562517.2012.696541

Lu B and Zheng F (2022). Influence of experiential teaching and itinerary assessment on the improvement of key competencies of students. International Journal of Emerging Technologies in Learning, 17(2): 173-188. https://doi.org/10.3991/ijet.v17i02.29007

Maslow A (1974). A theory of human motivation. Lulu Press, Morrisville, USA.

Mazur E (2017). Peer instruction. In: Kurz G and Harten U (Eds.), Peer instruction: 9-19. Springer Spektrum, Berlin, Germany. https://doi.org/10.1007/978-3-662-54377-1

McLeod S (2007). Maslow's hierarchy of needs. Simply Psychology. Available online at: https://www.simplypsychology.org/maslow.html

Mohorovicic S and Strcic V (2011). An overview of computer programming teaching methods. In the Central European Conference on Information and Intelligent Systems, Varazdin, Croatia: 47-52.

Papp-Varga Z, Szlávi P, and Zsakó L (2008). ICT Teaching methods–programming languages. Annales Mathematicae et Informaticae, 35(1): 163-172.

Rogers CR, Stevens B, Gendlin ET, Shlien JM, and Van Dusen W (1967). Person to person: The problem of being human: A new trend in psychology. Real People Press, Moab, USA.

Saito D and Yamaura T (2013). A new approach to programming language education for beginners with top-down learning. In the IEEE International Conference on Teaching, Assessment and Learning for Engineering, IEEE, Bali, Indonesia: 752-755. https://doi.org/10.1109/TALE.2013.6654538

Sangin M, Molinari G, Nüssli MA, and Dillenbourg P (2011). Facilitating peer knowledge modeling: Effects of a knowledge awareness tool on collaborative learning outcomes and processes. Computers in Human Behavior, 27(3): 1059-1067. https://doi.org/10.1016/j.chb.2010.05.032

Smart KL, Witt C, and Scott JP (2012). Toward learner-centered teaching: An inductive approach. Business Communication Quarterly, 75(4): 392-403. https://doi.org/10.1177/1080569912459752

Tan PH, Ting CY, and Ling SW (2009). Learning difficulties in programming courses: Undergraduates' perspective and perception. In the International Conference on Computer Technology and Development, IEEE, Kota Kinabalu, Malaysia: 42-46. https://doi.org/10.1109/ICCTD.2009.188

Tom M (2015). Five C framework: A student-centered approach for teaching programming courses to students with diverse disciplinary background. Journal of Learning Design, 8(1): 21-27. https://doi.org/10.5204/jld.v8i1.193

Wang L (2022). Influence of teacher behaviors on student activities in information-based classroom teaching. International Journal of Emerging Technologies in Learning, 17(2): 19-31. https://doi.org/10.3991/ijet.v17i02.28271

Welkener MM (2013). Understanding the complexities of cognition and creativity to reform higher education practice. International Journal of Educational Reform, 22(3): 221-234. https://doi.org/10.1177/105678791302200302

Wright GB (2011). Student-centered learning in higher education. International Journal of Teaching and Learning in Higher Education, 23(1): 92-97.

Xu Z, Ritzhaupt AD, Umapathy K, Ning Y, and Tsai CC (2021). Exploring college students' conceptions of learning computer science: A draw-a-picture technique study. Computer Science Education, 31(1): 60-82. https://doi.org/10.1080/08993408.2020.1783155