IMPROVING EDUCATIONAL PERFORMANCE BY TEACHING COMPUTATIONAL THINKING TO HIGHER EDUCATION STUDENTS

Husanova Surayyo Hama qizi
Karshi State University, teacher

Annotation: Explores the theoretical basis for improving the effectiveness of education by TEACHING computational thinking to HIGHER EDUCATION STUDENTS. Computational thinking is an important skill for students in today's digital age as it gives them the ability to analyze and solve problems in various fields.

Keywords: computational thinking, pedagogical strategies, models and structures, algorithms, data structures, abstraction, collaborative learning, project-based learning.

Аннотация: Исследует теоретические основы повышения эффективности образования путем обучения вычислительному мышлению студентов высших образований. Вычислительное мышление является важным навыком для студентов в современную цифровую эпоху, поскольку оно дает им возможность анализировать и решать проблемы в различных областях.

Ключевые слова: вычислительное мышление, педагогические стратегии, модели и структуры, алгоритмы, структуры данных, абстракция, совместное обучение, проектное обучение.

Computational thinking (CT) has emerged as a crucial skill in the digital age, as it equips individuals with the ability to analyze and solve problems in various domains. CT is the thought process involved in formulating and solving problems so that the solution can be represented in a form that can be executed by a computer. CT is not just about programming, but rather about a set of cognitive skills that enable individuals to understand how computers can help solve problems in their domains. CT is an essential skill for students in higher education institutions, as it provides them with the ability to analyze complex problems and develop efficient solutions using technology.

Teaching CT to students in higher education institutions is a complex task that requires a sound theoretical foundation and effective pedagogical strategies. In recent years, there has been an increasing emphasis on CT education in higher education institutions, as employers recognize the importance of this skill in the workplace. CT education can be viewed as a continuum, ranging from basic concepts of programming to more advanced skills, such as algorithmic thinking and data analysis. The goal of CT education is to develop students' abilities to think computationally, which involves problem-solving, abstraction, and algorithmic thinking.

Theoretical Foundations of Computational Thinking. CT is a multifaceted concept that encompasses several key components, such as algorithms, data structures, and abstraction. Algorithms are a set of instructions for solving a problem, while data structures are ways of organizing and storing data so that it can be efficiently accessed and manipulated. Abstraction involves simplifying complex problems by focusing on the essential details and ignoring irrelevant information.

Algorithms are a fundamental concept in CT education, as they provide a structured approach to problem-solving. Algorithms can be represented using flowcharts, pseudocode, or programming languages, and they can be applied to a variety of domains, such as mathematics, science, and engineering. By teaching students how to design and implement algorithms, educators can help them develop their problem-solving skills and enable them to tackle complex problems in their respective domains.

Abstraction is a key concept in CT education, as it involves simplifying complex problems by focusing on the essential details and ignoring irrelevant information. By teaching students how to abstract problems, educators can help them develop their critical thinking skills and enable them to tackle complex problems in a structured and organized manner.

Models and Frameworks for Teaching Computational Thinking. There are several models and frameworks for teaching CT, each with its own strengths and weaknesses. One such framework is the Computational Thinking for Everyone (CT4E) framework, which emphasizes the importance of problem-solving, abstraction, and algorithmic thinking in CT education. The CT4E framework is based on the following five components: problem decomposition, pattern recognition, abstraction, algorithms, and evaluation. By focusing on these components, educators can help students develop their CT skills in a structured and organized manner[1].

Another framework for teaching CT is the Computational Thinking Education Framework (CTEF), which emphasizes the importance of CT education across different domains and disciplines. The CTEF is based on the following five components: CT concepts and skills, CT teaching and learning, CT assessment, CT curriculum, and CT teacher education. By focusing on these components, educators can ensure that CT education is integrated into the broader curriculum and that students are exposed to CT concepts and skills in various domains[3].

Analysis and results. As mentioned in the related research, there are several effective pedagogical strategies for teaching computational thinking (CT) to students in higher education institutions. These include scaffolding, collaborative learning, and project-based learning.

Scaffolding involves breaking down complex problems into smaller, more manageable parts and providing students with step-by-step instructions to help them develop their CT skills. This strategy is effective because it provides students with the necessary support and guidance to help them learn at their own pace. Collaborative learning, on the other hand, allows students to work together and learn from each other. This strategy is effective because it allows students to develop their problem-solving and communication skills, as well as their ability to work in teams.

Project-based learning is also an effective strategy for teaching CT, as it allows students to apply their CT skills to real-world problems. This strategy involves providing students with a project that requires them to use their CT skills to solve a problem or develop a solution. By engaging in project-based learning, students can develop their CT skills in a meaningful and engaging way, as well as develop their ability to work independently and manage complex projects.

In terms of the theoretical foundations of teaching CT, there are several models and frameworks that can be used. For example, the Computational Thinking Framework by Wing (2006) outlines the key components of CT, including algorithms, data structures, and abstraction. This framework can be used to design CT curricula and to ensure that students are exposed to CT concepts and skills in various domains[4].

The related research also highlights some challenges to teaching CT, such as the need for faculty development and the lack of standardized assessments for CT skills. To overcome these challenges, higher education institutions may need to invest in faculty development programs that provide educators with the necessary knowledge and skills to teach CT effectively. Institutions may also need to develop standardized assessments for CT skills to ensure that students are acquiring the necessary skills and knowledge.

The analysis and results suggest that teaching CT to students in higher education institutions requires a sound theoretical foundation and effective pedagogical strategies. By incorporating scaffolding, collaborative learning, and project-based learning into CT instruction, educators can help students develop their CT skills in a structured and engaging manner, preparing them for success in the digital age. However, there are also challenges to teaching CT that must be addressed to ensure the effective integration of CT education into the higher education curriculum.

References:

1. Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: what is involved and what is the role of the computer science education community? ACM Inroads, 2(1), 48-54.

2. Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. Educational Researcher, 42(1), 38-43.
3. Guzdial, M. (2008). Education: Paving the way for computational thinking. Communications of the ACM, 51(8), 25-27.
4. Husanova S. H. OLIY TA'LIM MUASSASASI TALABALARINING DASTURLASH KO'NIKMALARINI SHAKLLANTIRISH (C++ DASTURLASH TILI MISOLIDA) //Экономика и социум. – 2023. – №. 2 (105). – С. 1360-1366.
5. Husanova S. THEORETICAL FOUNDATIONS OF TEACHING COMPUTATIONAL THINKING TO STUDENTS OF HIGHER EDUCATION INSTITUTIONS //International Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023.
6. Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.