Journal of Information Technology Education: Innovations in Practice

An Official Publication of the Informing Science Institute
InformingScience.org

JITEiip.org

Volume 23, 2024

CRITICAL THINKING: THE CODE TO CRACK COMPUTER SCIENCE EDUCATION

Sam Maesschalck — Nexova Group, Redu, Belgium & Lancaster University, Lancaster, UK — s.maesschalck@nexovagroup.eu

ABSTRACT

Aim/Purpose	This paper explores the potential value of critical thinking in computer science education and discusses strategies for its integration across the curriculum.
Background	As technology rapidly evolves and becomes increasingly integrated into society, there is a growing need for computer science graduates who can think critically about the ethical, societal, and technical implications of their work.
Methodology	This study employs a conceptual analysis approach, reviewing existing literature on critical thinking in computer science education and synthesising insights from various sources. The analysis focuses on identifying challenges in implementing critical thinking instruction and proposing practical solutions.
Contribution	This paper provides an overview of the current discourse on integrating critical thinking into computer science curricula. It explores the distinction between critical thinking and computational thinking, discusses various pedagogical approaches, and offers insights into potential challenges of implementation.
Findings	The paper identifies six key challenges in embedding critical thinking into computer science education. It proposes initial steps to address these challenges, including curriculum redesign, innovative assessment methods, and faculty development strategies.
Recommendations for Practitioners	Educators should adopt a balanced approach that complements technical education with critical thinking exercises, using diverse teaching methods such as dialogue-based teaching and authentic instruction.
Recommendations for Researchers	Future research should focus on empirical studies to assess the effectiveness of the proposed solutions and develop standardised tools for evaluating critical thinking skills in computer science.

Accepting Editor Stamatis Papadakis │ Received: June 22, 2024 │ Revised: September 14, September 28, October 2, 2024 │ Accepted: October 16, 2024.
Cite as: Maesschalck, S. (2024). Critical thinking: The code to crack computer science education. *Journal of Information Technology Education: Innovations in Practice, 23,* Article 13. https://doi.org/10.28945/5387

(CC BY-NC 4.0) This article is licensed to you under a Creative Commons Attribution-NonCommercial 4.0 International License. When you copy and redistribute this paper in full or in part, you need to provide proper attribution to it to ensure that others can later locate this work (and to ensure that others do not accuse you of plagiarism). You may (and we encourage you to) adapt, remix, transform, and build upon the material for any non-commercial purposes. This license does not permit you to use this material for commercial purposes.

Impact on Society	By enhancing critical thinking skills in computer science education, we can produce graduates who are better equipped to address complex technological challenges and their societal implications.
Future Research	Further investigation is needed into the most effective pedagogical approaches for teaching critical thinking in computer science, with a focus on multidisciplinary perspectives.
Keywords	computer science education, critical thinking, curriculum development, higher education, professional skills

INTRODUCTION

In academia, the significance of debate and intellectual inquiry stands paramount. However, a notable disconnect exists in imparting this critical thinking ethos to students within computer science education. This discrepancy becomes increasingly pertinent as technology embeds itself deeper into society and industry undergoes a rapid transformation. The pressing need for critical thinking in understanding the unique requirements of solutions within various sectors has to be better integrated into educational practices. While not unique to any single sector, these changes highlight the requirements to understand the political landscape and the intrinsic impact of new technologies more clearly than ever.

The problem addressed in this study is the undervaluation and inadequate integration of critical thinking skills in computer science education. This issue is significant because it impacts the ability of graduates to effectively address the complex challenges posed by rapidly evolving technologies and their societal implications. As technologies such as artificial intelligence, blockchain, and quantum computing advance, the need for computer scientists who can think critically about their work's broader implications becomes increasingly crucial (Khalil & Er, 2023).

Critical thinking in computer science education is not merely a supplementary skill but a crucial element in preparing students for innovative problem-solving and adaptation to evolving challenges. The importance of critical thinking in education has long been recognised (Fagin et al., 2006; Wing, 2006). However, the rapid advancement of technologies has dramatically altered the landscape of computer science, introducing new ethical dilemmas and societal implications that require robust critical thinking skills to navigate.

In today's technological environment, characterised by algorithmic bias, data privacy concerns, and the ethical use of AI, critical thinking skills are more pressing than ever. The current approach to computer science education often falls short in addressing these aspects despite their established importance in general educational literature. Therefore, it is argued that there is a need for a critical-thinking-centric approach in computer science education, one that recognises the diverse experiences of students, many of whom now enter programs with professional experience in software development or other areas, bringing unique insights and creating opportunities for peer learning (Hughes, 2019).

The objectives of this paper are:

1. To analyse the current state of critical thinking instruction in computer science education.
2. To explore the relationship between critical thinking skills and professional success in the technology sector
3. To identify key challenges in integrating critical thinking into computer science curricula.
4. To propose practical strategies for embedding critical thinking across the computer science educational experience.

The approach in this paper is a conceptual analysis drawing on existing literature in computer science education, critical thinking pedagogy, and insights from professional practice. This approach allows for a comprehensive examination of the issue, unifying theoretical frameworks with practical considerations. The analysis focuses on identifying challenges in implementing critical thinking instruction and proposing practical solutions based on empirical evidence and best practices in the field.

It is important to acknowledge the necessity of nurturing graduates who are not only technically proficient but also skilled in critical evaluation and ethical reasoning. These abilities are increasingly valued in the technology sector. By embedding critical thinking in the curriculum, students are equipped to effectively address real-world challenges, such as ethical issues in AI, cybersecurity threats, and strategic decision-making in software development. A study conducted in a university classroom setting demonstrated the positive impact of discussion-oriented and engaging activities on student academic performance and engagement (Abelha et al., 2020).

For educators, the relevance of this approach lies in its transformational potential for teaching methodologies. Integrating critical thinking enhances student engagement and learning outcomes. Current trends in computer science education, such as project-based learning and ethical tech development, underscore the timeliness of this thesis. Moreover, there is a obligation to create a clear alignment between the competencies taught in computer science courses and the industry's needs. Critical thinking goes beyond academic exercise and is a vital skill sought by employers in the tech industry.

The findings and recommendations presented here are intended to guide educators, curriculum designers, and policymakers in enhancing the critical thinking capabilities of future computer science professionals. By addressing these objectives, this research aims to start to bridge the gap between the technical focus of traditional computer science education and the broader cognitive skills required in professional settings.

This paper also argues for the inclusion of critical thinking as a fundamental component of computer science education, aiming to prepare graduates for the ever-evolving technological landscape, thereby enriching both their academic experience and professional trajectory. The subsequent sections of this paper will provide a comprehensive analysis of the integration of critical thinking in computer science education.

Critical Thinking and Computational Thinking

Within computer science education, two essential cognitive approaches stand out: critical thinking and computational thinking. Computational thinking has long been central to solving technical problems, while critical thinking introduces reflective reasoning and ethical evaluation, becoming increasingly important in today's complex technological landscape. Understanding the interplay between these modes of thought is crucial for developing well-rounded professionals. Those proficient in both cannot only solve technical challenges but can also assess the broader societal and ethical implications of their work. By exploring the definitions, similarities, differences, and implications of critical and computational thinking, educators can integrate these complementary skill sets into curricula, preparing students to navigate technical and societal challenges.

Defining Critical Thinking

Critical thinking is a multifaceted concept perceived differently across contexts and individuals (Lai, 2011). In some instances, it can be seen as disruptive – such as when a child questions a parent's directive. In academic settings, however, critical thinking is crucial, especially in higher education, where students are expected to engage with complex problems and challenge established norms.

Encouraging critical thinking is not always straightforward. For instance, contesting the decisions of a superior in a professional setting often requires years of experience. This complexity underscores the importance of integrating critical thinking into educational practices. Critical thinking can be either a source of enrichment or a challenge in the classroom, depending on the educator's approach. Some teachers may welcome the inquisitive student who asks probing questions, while others may find it disruptive.

Critical thinking becomes increasingly significant as students progress from undergraduate to graduate-level study. Initially, the focus is on absorbing pre-determined knowledge. However, students are expected to generate new ideas and work independently by the time they reach postgraduate studies, particularly at the PhD level. Fostering critical thinking at all stages of education ensures students develop the ability to question, analyse, and innovate. As such, critical thinking cannot be selectively applied but must be encouraged, in varying levels, throughout the educational journey.

To capture its diverse applications, various scholars have defined critical thinking differently, each emphasising a distinct facet of the concept:

- **Elder and Paul (2020):** "The art of thinking about thinking while thinking in order to make thinking better."
- **Ennis (1985):** "Reflective and reasonable thinking that is focused on deciding what to believe or do."
- **Halpern (1998):** "The use of those cognitive skills or strategies that increase the probability of a desirable outcome."
- **Sternberg (1986):** "The mental processes, strategies, and representations people use to solve problems, make decisions, and learn new concepts."

In education, critical thinking often encompasses open-mindedness, inquisitiveness, evaluative judgment, decision-making, and problem-solving (Lai, 2011). These qualities are vital in developing well-rounded, critically adept professionals. The challenge, however, lies in embedding critical thinking into teaching methodologies, ensuring it is translated into practice rather than remaining a theoretical exercise. Despite its benefits, implementing critical thinking in computer science education comes with challenges, which will be explored later in this paper.

DEFINING COMPUTATIONAL THINKING

Computational thinking (CT) is a structured approach to problem-solving grounded in computer science principles. Jeannette Wing (2006) defines it as "solving problems, designing systems, and understanding human behaviour by drawing on the concepts fundamental to computer science." More than just a technical skill, computational thinking fosters a mindset that allows individuals to break down complex problems, identify patterns, and create scalable, generalisable solutions.

The key components of computational thinking include:

- **Decomposition:** Breaking down a complex problem into smaller, manageable parts. This is particularly crucial in software development, where intricate systems are divided into modules that can be addressed individually before being integrated into a cohesive solution.
- **Pattern Recognition:** Identifying similarities or common elements across different problems. Recognising patterns allows students to apply previously learned solutions to new problems, increasing efficiency. This skill is essential when working with algorithms or debugging, as it enables students to identify recurring issues and resolve them quickly.
- **Abstraction:** Focusing only on relevant information and filtering out unnecessary details. Abstraction is key in programming, where it allows for the creation of general solutions that apply across various contexts without being bogged down by specifics.
- **Algorithmic Thinking:** Developing a step-by-step process to solve a problem. Algorithmic thinking underpins much of computer science, as algorithms are essential for executing tasks efficiently. Mastery of this skill encourages systematic thinking, ensuring that each step of the process is logically sound and efficient.

In computer science education, computational thinking equips students with a structured framework to approach problems methodically, mirroring how computers process information. Although it is often associated with coding, its principles are applicable across various domains, from mathematics and engineering to the social sciences.

While computational thinking offers a clear method for solving problems within technical frameworks, it shares some conceptual similarities with critical thinking. However, important distinctions remain between these two modes of thought. Exploring both their shared foundations and differences can reveal how they complement each other in educational contexts.

Similarities and Differences

Kules (2016) comprehensively compares critical and computational thinking, using the critical thinking framework by Elder and Paul (2020) and the Computer Science Teachers Association (CSTA) K-12 Computer Science Standards for computational thinking. Several key similarities exist between the two modes of thinking:

- **Concepts and Abstraction:** Both critical and computational thinking involve categorising and interpreting phenomena, often requiring abstraction to simplify complex problems
- **Formulation and Question-at-Issue:** Both approaches emphasise clearly identifying the question or problem at hand.
- **Data and Information:** Information and data are fundamental in both frameworks.
- **Systematic Analysis:** Both use systematic methods to tackle problems.
- **Logic:** Logical reasoning forms the foundation of both critical and computational thinking.

However, significant differences highlight the unique contributions of each:

- **Problem Framing:** Computational thinking focuses on framing problems for computer-based solutions, whereas critical thinking emphasises broader considerations, including purpose, point of view, and assumptions (Kules, 2016; Smith, 2021).
- **Algorithmic Thinking:** A core component of computational thinking, algorithmic thinking is not explicitly emphasised in critical thinking frameworks (Kules, 2016).
- **Evaluation:** Critical thinking places a strong emphasis on evaluating ideas and assumptions, which can be absent from computational thinking which may focus more on solution efficiency (Smith, 2021).
- **Scope:** Computational thinking is more limited in scope, focusing on problems that can be solved computationally, while critical thinking applies to a broader range of contexts (Kules, 2016; Smith, 2021).
- **Metacognition:** Critical thinking addresses metacognition more directly, encouraging individuals to reflect on their own thought processes. This is less prominent in computational thinking (Kules, 2016).

Rather than viewing computational thinking as a replacement for critical thinking, it is more productive to see them as complementary. Kules (2016) suggests that computational thinking enhances critical thinking, particularly in solving problems, making decisions, and interacting with technology and society. Smith (2021) similarly argues that computational thinking is a set of problem-solving tools that must be accompanied by critical thinking to fully address technological solutions' ethical, societal, and technical implications.

Implications for Computer Science Education

The complex relationship between computational thinking and critical thinking has several implications for computer science education:

- **Curriculum Design:** Educators should integrate both computational and critical thinking into curricula, ensuring students develop a balanced skill set. Understanding both modes of thought can enrich teaching and learning across disciplines (Kules, 2016).
- **Interdisciplinary Connections:** The relationship between these two types of thinking can support cross-disciplinary discourse, helping to link programming with broader institutional learning outcomes (Kules, 2016).
- **Contextual Application:** While computational thinking teaches technical problem-solving, critical thinking encourages students to evaluate the broader implications of their solutions, including ethical considerations.
- **Assessment Methods:** Educators must develop assessment methods that measure both computational and critical thinking skills, as traditional performance metrics may not adequately capture these competencies (Doleck et al., 2017).

In conclusion, while computational and critical thinking share some common elements, they are distinct yet complementary skills that bring unique value to computer science education. The challenge for educators lies in effectively integrating both into curricula and assessment methods, ensuring that students are equipped to navigate the complex landscape of modern technology and its societal impacts. Understanding and applying both modes of thinking can significantly enrich students' approaches to problem-solving and decision-making, fostering responsible and innovative professionals in the tech industry.

PROFESSIONAL SKILLS

While critical thinking is a vital skill in its own right, its importance is further amplified when intertwined with other professional skills in computer science education. The aim extends beyond imparting technical knowledge; it includes equipping students with a suite of professional skills crucial in today's workforce. These skills are often stimulated and complemented by critical thinking and are what employers increasingly seek in graduates.

Many institutions have dedicated career services teams that work with students on various levels to boost employability. However, these teams cannot do everything, nor does every student fully utilise the opportunities to interact with them. If we want to enhance students' employability, educators must also be actively involved in this process (Abelha et al., 2020).

The ability to communicate effectively, work as part of a team, solve problems creatively, and exhibit creativity are all essential in modern business environments, particularly in the technology sector. Consequently, educational approaches must reflect this reality. Traditionally, computer science curricula have focused heavily on technical skills. However, the need for graduates to articulate their knowledge and collaborate effectively with both technical and non-technical personnel is more pressing than ever. This realisation calls for a shift from a purely technical focus to one that also fosters broader professional skills, with critical thinking at its core. This aligns with findings from Pithers and Soden (2000), who argue that critical thinking is essential in preparing students for professional work in rapidly changing technological environments.

In many institutions, efforts are made to develop these skills through team-based projects and problem-solving assignments. However, these are not always successful in achieving their intended outcomes. Students may struggle with group dynamics or lack the motivation to engage fully with the task. This presents an opportunity for educators to guide students in navigating these challenges, thus aiding their development of crucial professional skills, including critical thinking. For example, addressing common issues such as unresponsiveness in group work not only resolves immediate problems but also prepares students for similar situations they may encounter professionally. The importance of developing these skills is further underscored by research showing that focusing on critical thinking has proven beneficial for students' performance in MBA programmes (D'Alessio et al., 2019), highlighting this skill's usefulness in professional settings.

While these approaches are valuable, they also highlight the potential drawback of increased workload for both students and educators. Designing, implementing, and assessing such activities requires significant time and effort, which must be balanced against other educational priorities. This challenge underscores the need for thoughtful curriculum design that integrates critical thinking and other professional skills without overwhelming the existing course structure.

Moreover, assignments in computer science often focus on specific technical requirements, such as using a particular programming language or software library. While this approach ensures consistency and simplifies grading, it may inadvertently limit students' creative and critical thinking. Allowing students some leeway to choose their tools or methods, provided they justify their choices, can foster deeper learning and critical engagement with the material. This approach aligns with the observations made by Abelha et al. (2020), highlighting the necessity for graduates to possess skills that extend beyond technical know-how.

To address these challenges, a redesign of assessment methods may be necessary. This could involve shifting the focus of marks towards professional skills, including critical thinking, rather than mainly evaluating system functionality in certain assignments. Such an approach would encourage students to develop and demonstrate their critical thinking abilities alongside their technical skills.

Ultimately, the goal is to graduate students who are not only proficient in the technical aspects of computer science but also skilled in communication, collaboration, and critical thinking. These skills are invaluable in professional settings where interdisciplinary teamwork and effective communication are the norm. By embedding these skills into the curricula, we enhance the employability of graduates and equip them to be more versatile and effective professionals. This is particularly important in IT, where technology changes rapidly, and many requirements must be balanced. Graduates need to be able to quickly adapt to these changes, keeping their skills relevant or adapting to the nuances of different sectors – abilities that are fundamentally rooted in critical thinking.

TECHNOLOGICAL ADVANCES

The rapid progression of technology, particularly in the realm of AI, underscores the crucial importance of critical thinking in computer science education. Recent research reveals a concerning gap in our approach to AI in education (AIEd), highlighting the urgent need for a more critical and pedagogically informed perspective.

Zawacki-Richter et al. (2019) conducted a systematic review of AI in education (AIEd) research, revealing a significant gap in critical reflection on the challenges and risks associated with its use in educational contexts. Their study highlighted that most of the research originates from computer science and STEM fields, with minimal contribution from educators. Additionally, they pointed out a weak link between existing research and established pedagogical theories. This is particularly concerning given the growing integration of AI technologies in educational environments.

The conclusions of Zawacki-Richter et al. (2019) reflect on the almost complete absence of critical examination of the challenges and risks of AIEd, the weak connection to theoretical and pedagogical perspectives, and the pressing need for further exploration of ethical and educational approaches in the application of AIEd in higher education. This gap in the research underscores a crucial need in computer science education: students must be taught to think critically about AI applications in education, considering not just their technical capabilities but also their pedagogical implications and potential risks.

When teaching about AI and other advanced technologies, engaging students in critical analysis of their potential positive and negative impacts is essential. This includes questioning the pedagogical foundations of AIEd applications and considering their ethical implications. For instance, how might an AI-driven tutoring system affect different learning styles or reinforce certain biases? What are the privacy implications of AI systems that collect and analyse student data?

Whittaker et al. (2021) further emphasise this need for critical thinking, discussing how AI systems are increasingly used to make sensitive determinations about individuals, including in areas related to disability. Computer science education must, therefore, include training in critically examining how these systems might encode and reproduce existing biases or problematic models of disability.

Consider, for example, proposed AI systems that claim to detect learning disabilities or predict academic performance. While these technologies may offer potential benefits, they also raise significant ethical concerns. Students must be taught to critically evaluate such systems, considering questions of privacy, consent, and the implications of automated assessments. This critical evaluation should extend to the datasets used to train these systems, the potential for cultural bias, and the risks of reinforcing stereotypes or exclusionary practices in education.

Furthermore, the development of AI systems often happens without meaningful input from diverse communities, including educators and students with diverse needs. This highlights the need for computer science education to emphasise inclusive design practices and critical thinking about representation in technology development. Students should be encouraged to question who is involved in the design process of AIEd tools and whose perspectives might be missing.

In conclusion, as technological advances continue to reshape education, the need for critical thinking skills becomes ever more apparent. Educators in computer science must rise to the challenge of not only teaching technical skills but also fostering the critical thinking abilities necessary to navigate the ethical, societal, and pedagogical implications of emerging technologies in education. This includes:

- critically evaluating the design and impact of new technologies, especially AI systems in education;
- analysing potential biases and exclusionary practices in AIEd design;
- considering diverse perspectives and promoting inclusive design practices in educational technology;
- assessing the ethical implications of AI implementations in educational settings;
- questioning the pedagogical foundations and effectiveness of AIEd tools; and
- encouraging interdisciplinary collaboration between computer scientists and education experts.

By emphasising these critical thinking skills, we prepare our students to become not just proficient practitioners but conscientious contributors to the field of computer science and education. They will be equipped to design, develop, and deploy more inclusive, ethical, and pedagogically sound technologies, benefiting all learners in our increasingly digital educational landscape.

INCORPORATING CRITICAL THINKING IN COMPUTER SCIENCE EDUCATION

In the landscape of computer science education, the integration of critical thinking is increasingly recognised as a cornerstone for developing a new generation of technologically adept and intellectually agile graduates. This section examines practical methods and strategies for embedding critical thinking within computer science curricula, synthesising current research and educational practices.

The approach taken in this analysis involves investigating literature in computer science education, critical thinking pedagogy, and insights from professional practice. This conceptual analysis allows for an examination of the issue, synthesising theory with practical considerations. The focus is on identifying challenges in implementing critical thinking instruction and proposing practical solutions based on empirical evidence and best practices in the field.

Two critical aspects of incorporating critical thinking in computer science education are explored – pedagogical approaches and practical application strategies. Each of these areas presents unique opportunities and challenges for educators seeking to enhance critical thinking skills in their students.

Pedagogical Approaches to Teaching Critical Thinking

Research has consistently demonstrated that critical thinking skills can be cultivated through deliberate educational practices (Abrami et al., 2015). Several pedagogical approaches have been identified as particularly effective in the context of computer science education:

1. *Dialogue-based Teaching and Mentorship:* Fostering an environment where inquiry is encouraged – through teacher-led discussions, student-initiated questioning, and group debates – promotes critical thinking by encouraging deeper analysis and reflection. Mentorship further enhances this process by providing personalised feedback and guidance, especially when integrated with dialogue-based and interactive learning activities (Dollinger et al., 2018; Micari & Calkins, 2021). Subcategories of this approach include:
   - teacher-posed questions
   - student-initiated questioning
   - whole-class discussions
   - group discussions
2. *Authentic or Anchored Instruction:* Applied problem-solving, real-world case studies, and simulations engage students in practical decision-making and help them relate technical knowledge to broader societal and ethical challenges. These methods enhance the ability to think critically about complex, real-world scenarios, aligning well with the practical nature of computer science education (Zainuddin et al., 2020). It involves techniques such as:
   - applied problem-solving
   - case studies
   - simulations
   - role-play

   A study by Zainuddin et al. (2020) demonstrated numerous benefits of gamification in teaching, which can be facilitated through technologies like Virtual Reality and Augmented Reality.

It is important to note that while these methods have shown effectiveness, they may encounter resistance from students accustomed to more traditional, lecture-based approaches. Overcoming this resistance requires patience and careful implementation on the part of educators.

Interestingly, while valuable, mentoring has been found to be less effective as a standalone method for teaching critical thinking. However, mentoring can significantly enhance the learning process when combined with dialogue-based teaching and authentic instruction. This implies that the role of educators in mentoring should be strategically aligned with these more dynamic and interactive teaching methods.

Applying Critical Thinking to Computer Science Education

Integrating critical thinking into computer science education requires more than just isolated strategies; it demands a holistic rethinking of how computer science is taught. Critical thinking empowers students to assess, evaluate, and address the complexities of technological challenges they will face in their careers. In this section, we explore practical methods for embedding critical thinking into the computer science curriculum and address the inherent challenges.

Fostering a culture of inquiry

One of the foundational ways to integrate critical thinking is by fostering a culture of inquiry in the classroom. Asking questions like "Why?" or "What if?" challenges students to move beyond rote learning and consider the deeper implications of their technical decisions. Encouraging students to question assumptions engages them in a more critical, reflective learning process, which is crucial for problem-solving and decision-making in computer science. This has been found to be refreshing and enjoyable by students (Maesschalck et al., 2023).

Open-ended assignments and flexibility

Open-ended assignments provide an excellent avenue for students to engage in critical thinking. Allowing them to choose their tools, programming languages, or approaches to a problem encourages creative problem-solving while demanding that they justify their decisions. This challenges them academically and mirrors the type of critical decision-making they will encounter in their professional lives. Studies have shown that giving students flexibility in assignments leads to enhanced engagement and improved learning outcomes (Vrieler & Salminen-Karlsson, 2022).

However, this flexibility poses a challenge in assessment. When students take different approaches to solving a problem, ensuring that evaluation remains consistent and objective becomes more difficult. Careful development of grading rubrics that focus on critical reasoning, problem-solving processes, and the justification of choices, rather than just the technical outcome, can help mitigate potential bias in assessment.

Integrating AI tools in critical thinking exercises

The rapid rise of AI tools, such as ChatGPT, presents a unique opportunity to integrate critical thinking into the curriculum. Rather than viewing these tools as threats to academic integrity, educators can use them to foster critical engagement. For instance, students can critically evaluate AI-generated code, analyse the limitations of such tools, or use them as collaborative partners in programming exercises, which has been shown to improve their analytical skills (Singh et al., 2023). This approach prepares students for the AI-driven landscape of their future careers and deepens their understanding of the ethical implications and potential biases inherent in AI technologies.

Co-creation of learning materials and assessments

Another effective strategy is involving students in the co-creation of their learning materials and assessments. By actively participating in creating coursework, students take ownership of their learning and develop a more reflective, critical approach. Co-creation fosters critical thinking because students must critically assess their peers' and their own work, consider alternative perspectives, and justify their decisions. This method also encourages inclusivity by drawing on diverse student perspectives to enrich the learning environment (Dollinger et al., 2018).

The challenge of assessing critical thinking

One of the primary challenges in integrating critical thinking into computer science education is assessment. Traditional grading methods often focus on measurable technical skills, but critical thinking involves more nuanced skills such as reasoning, ethical reflection, and decision-making, which can be harder to evaluate objectively. There is also the risk of bias in assessment, as the evaluation of critical thinking can be subjective. This is particularly true when grading tasks like open-ended assignments or peer reviews, where the quality of reasoning and creativity may be interpreted differently by each educator.

To address this, educators can use rubrics that clearly define the criteria for assessing critical thinking, such as the depth of analysis, justification of decisions, and ability to consider ethical implications. These rubrics should be transparent and shared with students, ensuring they understand how their critical thinking will be evaluated. Additionally, incorporating peer review and self-assessment can add multiple perspectives to the evaluation process, reducing the reliance on a single source of judgment and thus mitigating potential bias (Topping, 1998).

Alignment with Pedagogical Principles

The strategies proposed for incorporating critical thinking into computer science education are grounded in well-established pedagogical principles, ensuring their effectiveness and theoretical soundness.

Constructivism: The dialogue-based teaching and authentic instruction methods align with constructivist learning theory (Piaget, 1976; Vygotsky & Cole, 1978). These approaches allow students to actively construct their understanding of critical thinking within computer science contexts rather than passively receiving information.

Active Learning: Many of the suggested approaches, such as ethical case studies and design critique sessions, exemplify active learning principles (Bonwell & Eison, 1991). Engaging students directly in the learning process promotes critical engagement and retention of information. Research has shown that active learning methods can improve student outcomes, particularly in developing problem-solving and critical-thinking abilities (Prince, 2004).

Bloom's Taxonomy: The proposed assessment methods correspond to higher-order thinking skills in Bloom's Taxonomy (Anderson & Krathwohl, 2001). For instance, code review exercises align with the "Evaluate" level, while open-ended problem solving corresponds to the "Create" level, ensuring a comprehensive development of critical thinking abilities.

Experiential Learning: The incorporation of real-world scenarios and industry-relevant projects is rooted in Kolb's (2014) Experiential Learning Theory. This approach allows students to cycle through concrete experience, reflective observation, abstract conceptualisation, and active experimentation, enhancing their critical thinking skills through practical application.

Scaffolding**:** The gradual implementation of critical thinking exercises aligns with the principle of scaffolding (Wood et al., 1976). This approach provides students with necessary support as they develop more complex thinking skills, gradually reducing assistance as competence increases.

Metacognition: The use of reflective journals promotes metacognition (Flavell, 1979), enhancing students' awareness of their own thinking processes and fostering self-regulated learning. These exercises help students become more aware of their problem-solving approaches and identify areas for improvement, fostering self-regulated learning and critical reflection. Empirical research supports using metacognitive activities to enhance academic performance and critical thinking (Tanner, 2012).

Social Learning Theory: Group discussions and peer evaluations leverage Vygotsky and Cole's (1978) Social Learning Theory, utilising social interaction to enhance learning and critical thinking development. By working collaboratively, students engage in shared learning experiences that challenge them to think critically about both their own ideas and the ideas of others. Fostering a deeper understanding of the material.

Problem-Based Learning: Many of the suggested assessment methods, particularly those involving real-world ethical dilemmas and open-ended problems, align with problem-based learning principles (Barrows & Tamblyn, 1980) as a subset of Active Learning. This approach enhances critical thinking by engaging students with complex, authentic problems. Research has shown that PBL fosters independent learning, problem-solving, and the ability to apply knowledge in practical situations (Hmelo-Silver, 2004), which are essential skills in computer science education.

By grounding our approach in these established pedagogical principles, we ensure that the integration of critical thinking into computer science education is not only innovative but also theoretically sound and empirically supported. This foundation provides a robust framework for educators to implement these strategies effectively, fostering the development of critical thinking skills that are crucial for future computer science professionals.

CHALLENGES OF INCORPORATING CRITICAL THINKING INTO COMPUTER SCIENCE EDUCATION

Incorporating critical thinking into computer science education presents several interconnected challenges. While critical thinking is essential for preparing students to tackle complex technological issues, educators face significant obstacles in terms of teaching style shifts, student resistance, assessment methods, technological complexity, and institutional support.

THE PRIMARY CHALLENGES

Incorporating critical thinking into computer science education introduces several key challenges. Many educators are accustomed to traditional methods that focus primarily on technical skills. Transitioning to a model that prioritises reflective reasoning, ethical consideration, and cognitive engagement is not straightforward.

A significant challenge is the shift in teaching styles. Many educators in the field rely on didactic, lecture-based methods, and the move towards inquiry-based, discussion-led approaches can feel unfamiliar and intimidating (Pithers & Soden, 2000). These new methods demand the creation of an environment where ethical dilemmas, reflective reasoning, and open questioning become integral to the learning process – skills for which many educators lack formal training (Janssen et al., 2019). Without sufficient training, teachers may struggle to effectively integrate critical thinking activities into their courses (Zwiers & Crawford, 2023).

Another key challenge is resistance from students. Many students, accustomed to structured and objective tasks, may struggle with assignments that do not have clear answers. Gokhale (1995) noted that students often favour tasks with definitive solutions, which can lead to disengagement when faced with more open-ended or ambiguous problems. This resistance is compounded by the perception that critical thinking exercises divert from the technical skills essential for success in the field, potentially lowering student motivation and performance (Prince, 2004).

Assessing critical thinking also presents challenges. Traditional assessments in computer science are often objective and focus on program functionality. However, critical thinking requires the evaluation of reasoning, ethical judgment, and the ability to synthesise diverse perspectives (Nicol & Macfarlane-Dick, 2006). Designing fair and consistent rubrics for these assessments remains a challenge, especially in larger classes where objective grading is more straightforward (Topping, 1998).

Lastly, a lack of institutional support exacerbates these challenges. Many institutions focus on meeting industry demands for technical skills, leaving limited time and resources for professional development that could help educators shift to more reflective teaching practices (Kolb, 2014). Without sufficient support, educators may struggle to implement these changes effectively despite the potential benefits for students' long-term development.

POTENTIAL DRAWBACKS OF THE CRITICAL THINKING APPROACH

While integrating critical thinking into computer science education offers numerous benefits, it also comes with potential drawbacks that educators must navigate carefully. These drawbacks revolve around finding a balance between developing critical thinking skills, maintaining technical proficiency, and managing the practical implications of pedagogical shifts.

One major drawback is the risk of reduced focus on technical skills. As more time is allocated to reflective tasks, students may spend less time mastering essential technical competencies, such as coding and algorithm development. In a field where technical expertise is the primary criterion for entry-level jobs, this shift could leave students feeling underprepared (Smith, 2021). Striking a balance between fostering critical thinking and ensuring technical skill development is crucial but difficult.

Another concern is the increased workload for both students and educators. Critical thinking activities – such as open-ended problem-solving, ethical debates, and reflective journaling – are time-intensive and require significant cognitive effort from students. For educators, these activities necessitate more extensive grading and feedback, increasing the workload compared to traditional technical assessments and potentially leading to burnout (Janssen et al., 2019).

A related drawback is the challenge of subjective assessment. Technical skills can be measured objectively, but evaluating critical thinking involves more subjective judgement of reasoning, ethical reflection, and synthesis of ideas. This variability in assessment, particularly in large classes where multiple graders may be involved, can lead to inconsistencies that frustrate students who may perceive grading as unfair (Topping, 1998).

Student resistance is another significant issue. Many students accustomed to clearly defined, objective tasks may struggle with open-ended assignments that require critical analysis and reflection. Ambiguity in such tasks can lead to disengagement, especially for those who thrive in structured learning environments (Prince, 2004). Introducing critical thinking gradually and setting clear expectations can help mitigate this resistance.

Lastly, there is the risk of misalignment with industry expectations. While critical thinking is highly valued in leadership and innovation roles, many entry-level positions in computer science demand strong technical skills. Graduates who have focused heavily on critical thinking may be disadvantaged when competing for jobs that prioritise technical proficiency (Singh et al., 2023). This raises concerns about how best to balance immediate job readiness with the long-term benefits of developing critical thinking skills.

The Six Challenges

Addressing the integration of critical thinking in computer science education involves confronting several challenges, as discussed previously. These challenges have to be overcome to achieve a successful incorporation of critical thinking skills. Here, I outline six high-level challenges that flow out of the challenges and drawbacks discussed, combined with an initial step toward solving them.

1. **Limited Resources and Training**

   Challenge: A significant barrier is the lack of training or resources available to educators for effectively teaching critical thinking in a computer science context. Educators often need to balance technical instruction with fostering critical engagement, which requires new pedagogical approaches.

   Solution: Professional development programs focused on teaching critical thinking should be prioritised. These programs could include workshops and online courses to train educators in integrating critical thinking into their technical instruction. Collaborations with industry experts can also enhance training by bringing real-world insights into the classroom. Research by Janssen et al. (2019) highlights the positive impact of training higher education teachers in critical thinking methods, showing improvements in both teaching practices and student engagement.

2. **Curriculum Overhaul**

   Challenge: Fitting critical thinking into an already dense computer science curriculum is a daunting task. Many programs are heavily focused on technical skills, leaving little room for reflective reasoning or ethical discussions.

   Solution: A modular approach to curriculum design can help. Students can practice critical thinking without detracting from core technical skills by embedding critical thinking exercises within existing technical lessons – such as ethical case studies during programming or algorithm design. For example, when teaching algorithms, educators can introduce discussions about algorithmic bias and its societal impact. Studies show that embedding ethics in technical education improves both critical engagement and technical mastery (Doleck et al., 2017).

3. **Measuring Success**

   Challenge: Assessing critical thinking is more subjective than technical skills, and traditional assessment methods may not adequately capture students' critical engagement. There is a risk that grading can become inconsistent or biased, depending on the educator's interpretation.

   Solution: A multifaceted assessment strategy can address this issue. Rubrics that clearly define critical thinking criteria – such as depth of analysis, reasoning, and ethical considerations – can help standardise assessments. Peer review and self-assessment can also encourage reflection and reduce instructor bias. Studies such as Topping (1998) have demonstrated that peer assessment fosters critical thinking by allowing students to evaluate each other's reasoning and approach.

4. **Student Resistance**

   Challenge: Some students may resist the shift toward a critical thinking-focused approach, particularly when it deviates significantly from traditional, more straightforward teaching methods. This resistance can stem from unfamiliarity with open-ended tasks that don't have clear, objective answers.

   Solution: Gradual integration of critical thinking exercises can help ease this transition. Starting with simpler tasks and progressively building to more complex ones allows students to become more comfortable with ambiguity. Additionally, incorporating gamification techniques, such as coding challenges or real-world problem-solving exercises like hackathons, can engage students more effectively. Research by Prince (2004*)* supports the use of active learning techniques to increase student motivation and foster critical engagement.

5. **Keeping Pace with Technology**

   Challenge: The fast pace of technological advancement in computer science means that curricula must be constantly updated to stay relevant. Integrating critical thinking into this rapidly evolving landscape can be difficult, especially when new technologies like artificial intelligence introduce additional layers of complexity.

   Solution: Establishing advisory boards consisting of industry experts, alumni, and academic researchers can help institutions keep their curricula current with emerging technologies. Additionally, educators should incorporate tools like AI into their teaching, allowing students to critically evaluate AI-generated solutions or reflect on the ethical implications of using AI in practice. For example, tasks where students critically analyse AI-generated code can help develop both technical and critical thinking skills. Research by Singh et al. (2023) supports the use of AI tools in education to foster deeper critical engagement.

6. **Diverse Student Backgrounds**

   Challenge: Computer science students come from a wide range of cultural and educational backgrounds, leading to different levels of familiarity with critical thinking exercises. Some students may struggle to engage with critical thinking tasks due to a lack of experience with reflective reasoning or open-ended problem-solving.

   Solution: Culturally responsive teaching practices can help ensure that all students benefit from critical thinking instruction. This includes offering diverse case studies and real-world examples that reflect various global contexts. Peer collaboration can also help bridge gaps in student experience by fostering discussions where students learn from each other's perspectives. Educators can create a more inclusive learning environment by incorporating differentiated instruction and diverse assessment methods. Research on culturally responsive teaching has shown positive effects on student engagement and learning outcomes, particularly in diverse classrooms (Zwiers & Crawford, 2023).

Future research in this area should focus on further empirical studies to assess the effectiveness of these solutions, innovations in pedagogy, the development of standardised tools for evaluating critical thinking skills, and the impact of cultural and educational diversity on critical thinking skill acquisition. Additionally, the role of industry-academia partnerships in keeping curricula aligned with current trends and technological advancements is crucial. Addressing these challenges effectively is key to successfully integrating critical thinking into computer science education, enhancing the quality and relevance of the education provided. These efforts are vital if we want to develop a workforce equipped with critical analysis, problem-solving, and ethical decision-making skills, ready to face the complex challenges of the technological world.

ASSESSING CRITICAL THINKING IN COMPUTER SCIENCE

Assessing critical thinking in computer science education requires a multifaceted approach, using various methods that balance technical proficiency with reflective analytical skills. Below are several strategies that can be employed to evaluate students' critical thinking abilities in both practical and theoretical contexts.

1. **Code Review Exercises**

Method: Students critically analyse and provide feedback on code written by their peers or in fictional scenarios, focusing on areas such as efficiency, readability, and potential improvements. This method not only tests technical competence but also encourages students to think critically about software quality and development practices.

Rationale: This exercise aligns with the constructivist principle of learning through reflection and interaction. Students engage in deeper learning by considering alternative approaches and making reasoned judgments about coding practices. Research shows that peer review can be an effective tool for promoting critical thinking in collaborative learning environments (Topping, 1998).

2. **Ethical Case Studies**

Method: Present students with real-world ethical dilemmas in tech—such as data privacy issues or algorithmic bias—and assess their ability to analyse the problem, consider multiple perspectives, and propose ethically sound solutions.

Rationale: By incorporating ethical case studies, educators can leverage problem-based learning principles. This method encourages students to engage with authentic problems and develop critical thinking by evaluating both the technical and social implications of technology. The framework by Elder and Paul (2020) for critical thinking also emphasises the importance of evaluating decisions through a lens of purpose and ethical considerations.

3. **Design Critique Sessions**

Method: Students present software designs and architectures for peer and instructor critique. Feedback focuses on how well the design aligns with best practices and the thought process behind specific decisions.

Rationale: Design critique sessions emphasise experiential learning, allowing students to engage in critical self-reflection and peer evaluation. Kolb's (2014) experiential learning theory supports the idea that learning occurs most effectively when students actively apply theoretical knowledge in a practical context. Studies by McConnell (2002) suggest that peer evaluation in design work fosters deeper critical reflection, improving both technical skills and the ability to justify design decisions without the assistance of teaching staff.

4. **Algorithmic Bias Detection**

Method: Provide students with datasets and algorithms, asking them to critically analyse potential biases in the data and algorithm outcomes. This exercise develops both technical and ethical critical thinking skills, especially in fields like artificial intelligence.

Rationale: This method integrates critical analysis with technical problem-solving, encouraging students to think beyond technical implementation and consider broader societal implications. It reflects the growing importance of ethical AI development, as highlighted in recent studies (Singh et al., 2023).

5. **Open-Ended Problem Solving**

Method: Students are presented with complex, ill-defined problems and tasked with breaking them down and proposing solutions. Assessment focuses not only on the solution itself but also on how the students approach the problem.

Rationale: Open-ended problem-solving emphasises higher-order thinking in Bloom's Taxonomy, particularly at the "Analyse" and "Create" levels (Anderson & Krathwohl, 2001). This method challenges students to navigate ambiguity, requiring them to justify their decisions and reflect on their problem-solving process. The research by Hmelo-Silver (2004) on problem-based learning highlights how engaging with real-world, ill-defined problems develops critical thinking and problem-solving skills by involving students in iterative inquiry, analysis, and reflection.

6. **Reflective Journals**

Method: Students maintain journals where they reflect on their problem-solving processes, challenges faced, and lessons learned throughout a course.

Rationale: Reflective journals encourage metacognition, allowing students to critically analyse their own thinking. This method fosters self-regulation and deeper engagement with the material, as supported by Flavell's (1979) theory of metacognition.

7. **Peer Evaluation Rubrics**

Method: Develop detailed rubrics for peer code reviews that include criteria for assessing the depth and quality of critical analysis, encouraging students to think critically while evaluating their peers' work.

Rationale: Peer review not only helps distribute the workload but also fosters critical engagement with content from multiple perspectives. By clearly outlining assessment criteria, it mitigates potential biases, ensuring consistency and transparency (Topping, 1998).

These assessment methods should be combined to comprehensively evaluate students' critical thinking skills in computer science contexts. It's important to note that assessing critical thinking is an ongoing process that should be integrated throughout the curriculum rather than treated as a one-time evaluation. Therefore, the challenge of standardisation across different courses or institutions should also be highlighted. Ensuring consistency in the depth and quality of critical thinking assessment remains a significant hurdle.

Moreover, transparency in assessment criteria is crucial. Students should be made aware of how their critical thinking skills are being evaluated, which can encourage more thoughtful engagement with the material. Regular feedback on these assessments, highlighting areas of strength and opportunities for improvement in critical thinking, can further enhance the learning process. By incorporating these diverse assessment methods, educators can more effectively gauge and foster the development of critical thinking skills essential for future computer science professionals.

CONCLUSION

The integration of critical thinking into computer science education is essential for preparing students to navigate the complex ethical, social, and technical challenges they will encounter in their careers. As outlined in this paper, fostering critical thinking requires significant shifts in teaching approaches, assessment methods, and institutional support. From peer review and design critique sessions to ethical case studies and open-ended problem-solving tasks, a variety of strategies can be employed to help students engage in reflective, analytical thinking.

Despite the clear benefits, challenges such as student resistance, increased workload, and difficulty assessing open-ended tasks remain significant barriers. However, these challenges can be mitigated with well-designed rubrics, peer and self-assessment, and gradual integration of critical thinking into technical instruction.

Critical thinking enhances students' technical abilities and their capacity for ethical reasoning and innovation. As technology continues to evolve, the ability to think critically about its societal implications will be increasingly valued. Future research should focus on exploring how emerging technologies such as artificial intelligence and machine learning can be integrated into critical thinking exercises, allowing students to critically assess AI-generated solutions, data biases, and ethical concerns. Additionally, studies could examine the long-term impact of critical thinking on student outcomes, particularly in relation to employability and adaptability in rapidly changing technological environments.

There is also a need for further investigation into scalable methods for assessing critical thinking, particularly in large classroom settings. Innovative assessment tools, including AI-assisted grading systems, may offer solutions for evaluating open-ended, reflective tasks more efficiently. Moreover, longitudinal studies tracking how computer science students develop critical thinking skills over time could provide valuable insights into the most effective pedagogical approaches.

In conclusion, embedding critical thinking within computer science education is not merely an option but a necessity for cultivating future professionals who can critically engage with technology and contribute to its responsible development. The future of computer science lies not only in technical expertise but in the ability to question, reflect, and innovate with purpose. Continued research and investment in critical thinking pedagogy will be crucial in shaping the next generation of technologists.

REFERENCES

Abelha, M., Fernandes, S., Mesquita, D., Seabra, F., & Ferreira-Oliveira, A. T. (2020). Graduate employability and competence development in higher education – A systematic literature review using PRISMA. *Sustainability*, *12*(15), 5900. https://doi.org/10.3390/su12155900

Abrami, P. C., Bernard, R. M., Borokhovski, E., Waddington, D. I., Wade, C. A., & Persson, T. (2015). Strategies for teaching students to think critically: A meta-analysis. *Review of Educational Research*, *85*(2), 275-314. https://doi.org/10.3102/0034654314551063

Anderson, L. W., & Krathwohl, D. R. (2001). *A taxonomy for learning, teaching, and assessing: A revision of Bloom's taxonomy of educational objectives*. Longman.

Barrows, H. S., & Tamblyn, R. M. (1980). *Problem-based learning: An approach to medical education*. Springer.

Bonwell, C. C., & Eison, J. A. (1991). *Active learning: Creating excitement in the classroom* (ED336049). ERIC.

D'Alessio, F. A., Avolio, B. E., & Charles, V. (2019). Studying the impact of critical thinking on the academic performance of executive MBA students. *Thinking Skills and Creativity*, *31*, 275-283. https://doi.org/10.1016/j.tsc.2019.02.002

Doleck, T., Bazelais, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: Exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, *4*, 355-369. https://doi.org/10.1007/s40692-017-0090-9

Dollinger, M., Lodge, J., & Coates, H. (2018). Co-creation in higher education: Towards a conceptual model. *Journal of Marketing for Higher Education*, *28*(2), 210-231. https://doi.org/10.1080/08841241.2018.1466756

Elder, L., & Paul, R. (2020). *Critical thinking: Learn the tools the best thinkers use.* Foundation for Critical Thinking.

Ennis, R. H. (1985). A logical basis for measuring critical thinking skills. *Educational Leadership*, *43*(2), 44-48.

Fagin, B., Harper, J., Baird, L., Hadfield, S., & Sward, R. (2006). Critical thinking and computer science: Implicit and explicit connections. *Journal of Computing Sciences in Colleges*, *21*(4), 171-177. https://dl.acm.org/doi/abs/10.5555/1127389.1127423

Flavell, J. H. (1979). Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry. *American Psychologist*, *34*(10), 906-911. https://doi.org/10.1037/0003-066X.34.10.906

Gokhale, A. A. (1995). Collaborative learning enhances critical thinking. *Journal of Technology Education*, *7*(1), 22-30. https://doi.org/10.21061/jte.v7i1.a.2

Halpern, D. F. (1998). Teaching critical thinking for transfer across domains: Disposition, skills, structure training, and metacognitive monitoring. *American Psychologist*, *53*(4), 449-455. https://doi.org/10.1037/0003-066X.53.4.449

Hmelo-Silver, C. E. (2004). Problem-based learning: What and how do students learn?. *Educational Psychology Review*, *16*, 235-266. https://doi.org/10.1023/B:EDPR.0000034022.16470.f3

Hughes, M. (2019). *A guide to statutory social work interventions: The lived experience.* Red Globe Press.

Janssen, E. M., Mainhard, T., Buisman, R. S., Verkoeijen, P. P., Heijltjes, A. E., van Peppen, L. M., & van Gog, T. (2019). Training higher education teachers' critical thinking and attitudes towards teaching it. *Contemporary Educational Psychology*, *58*, 310–322. https://doi.org/10.1016/j.cedpsych.2019.03.007

Khalil, M., & Er, E. (2023). Will ChatGPT get you caught? Rethinking of plagiarism detection. In P. Zaphiris, & A. Ioannou (Eds.), *Learning and collaboration technologies.* Springer. https://doi.org/10.1007/978-3-031-34411-4_32

Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development.* Prentice-Hall.

Kules, B. (2016). Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes. *Proceedings of the Association for Information Science and Technology*, *53*(1), 1-6. https://doi.org/10.1002/pra2.2016.14505301092

Lai, E. R. (2011). *Critical thinking: A literature review.* Pearson. https://www.academia.edu/download/34330969/CriticalThinkingReviewFINAL.pdf

Maesschalck, S., Bradbury, M., & Giotsas, V. (2023, May). Into the heat of the debate: Simulating a program committee within computer science education. *Proceedings of the IEEE Global Engineering Education Conference, Kuwait, Kuwait*, 1-5. https://doi.org/10.1109/EDUCON54358.2023.10125275

McConnell, D. (2002). The experience of collaborative assessment in e-learning. *Studies in Continuing Education*, *24*(1), 73-92. https://doi.org/10.1080/01580370220130459

Micari, M., & Calkins, S. (2021). Is it OK to ask? the impact of instructor openness to questions on student help-seeking and academic outcomes. *Active Learning in Higher Education*, *22*(2), 143–157. https://doi.org/10.1177/1469787419846620

Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. *Studies in Higher Education*, *31*(2), 199–218. https://doi.org/10.1080/03075070600572090

Piaget, J. (1976). Piaget's theory. In B. Inhelder, H. H. Chipman, & C. Zwingmann, C. (Eds.), *Piaget and his school: A reader in developmental psychology* (pp. 11-23). Springer. https://doi.org/10.1007/978-3-642-46323-5_2

Pithers, R. T., & Soden, R. (2000). Critical thinking in education: A review. *Educational Research*, *42*(3), 237-249. https://doi.org/10.1080/001318800440579

Prince, M. (2004). Does active learning work? A review of the research. *Journal of Engineering Education*, *93*(3), 223-231. https://doi.org/10.1002/j.2168-9830.2004.tb00809.x

Singh, H., Tayarani-Najaran, M.-H., & Yaqoob, M. (2023). Exploring computer science students' perception of chatgpt in higher education: A descriptive and correlation study. *Education Sciences*, *13*(9), 924. https://doi.org/10.3390/educsci13090924

Smith, J. M. (2021). Is computational thinking critical thinking? In Y. Wen, Y. Wu, G. Qi, S.-C. Guo, J. M. Spector, S. Chelliah, Kinshuk, & Y.-J. Lan (Eds.), *Expanding global horizons through technology enhanced language learning* (pp. 191-201). Springer. https://doi.org/10.1007/978-981-15-7579-2_11

Sternberg, R. J. (1986). *Critical thinking: Its nature, measurement, and improvement* (ED272882). ERIC https://eric.ed.gov/?id=ED272882

Tanner, K. D. (2012). Promoting student metacognition. *CBE – Life Sciences Education*, *11*(2), 113-120. https://doi.org/10.1187/cbe.12-03-0033

Topping, K. (1998). Peer assessment between students in colleges and universities. *Review of Educational Research*, *68*(3), 249-276. https://doi.org/10.3102/00346543068003249

Vrieler, T., & Salminen-Karlsson, M. (2022). A sociocultural perspective on computer science capital and its pedagogical implications in computer science education. *ACM Transactions on Computing Education*, *22*(4), 1–23. https://doi.org/10.1145/3487052

Vygotsky, L. S., & Cole, M. (1978). *Mind in society: The development of higher psychological processes.* Harvard University Press.

Whittaker, M., Alper, M., Bennett, C. L., Hendren, S., Kaziunas, L., Mills, M., Morris, M. R., Rankin, J., Rogers, E., Salas, M., & West, S. M. (2021). *Disability, bias, and AI.* AI Now Institute. https://ainowinsti-tute.org/wp-content/uploads/2023/04/disabilitybiasai-2019.pdf

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33–35. https://doi.org/10.1145/1118178.1118215

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, *17*(2), 89-100. https://doi.org/10.1111/j.1469-7610.1976.tb00381.x

Zainuddin, Z., Chu, S. K. W., Shujahat, M., & Perera, C. J. (2020). The impact of gamification on learning and instruction: A systematic review of empirical evidence. *Educational Research Review*, *30*, 100326. https://doi.org/10.1016/j.edurev.2020.100326

Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher education – where are the educators? *International Journal of Educational Technology in Higher Education*, *16*(1), 1-27. https://doi.org/10.1186/s41239-019-0171-0

Zwiers, J., & Crawford, M. (2023). *Academic conversations: Classroom talk that fosters critical thinking and content understandings.* Routledge. https://doi.org/10.4324/9781032680514

AUTHOR

Sam Maesschalck is a Security Training Engineer at Nexova Group, where he is responsible for the development and delivery of training for the European Space Agency's Cybersecurity Centre of Excellence. He is also a researcher at Lancaster University, where he obtained his PhD. Before, he was a Senior Teaching Associate at Lancaster University.

His main research interests lie in the security of critical infrastructure/OT systems and information technology education with a focus on cybersecurity.