What is Computing Education Research (CER)?

Mats Daniels, Lauri Malmi, Arnold Pears, and Simon

1 Introduction

What is Computing Education Research (CER), who is doing research in the field, and what characterizes different types of CER? We address these questions in this chapter from our perspectives as active researchers in the field for the last almost 30 years. It is not our purpose to act as gatekeepers but rather to provide views on CER and discuss how the research fields and community have evolved. The goal is to provide the reader with a historical perspective and a structure to understand CER, in which we hope the CER community will feel at home. One particular aspect of CER is change, which provides a challenge in framing this chapter. Changes in CER have been considerable, from its early history in computer science to the vast diversity of computing education at all educational levels as well as in non-formal settings of today. This change stems partly from far-reaching changes in the nature and scope of computing education over the past 50 years. Early efforts in computer science education in the middle of the twentieth century were restricted

M. Daniels (✉)
Uppsala University, Uppsala, Sweden
e-mail: mats.daniels@it.uu.se

L. Malmi
Aalto University, Espoo, Finland
e-mail: lauri.malmi@aalto.fi

A. Pears
KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: pears@kth.se

Simon
Wadalba, NSW, Australia
e-mail: pears@kth.se

M. Apiola et al. (eds.), *Past, Present and Future of Computing Education Research*,
https://doi.org/10.1007/978-3-031-25336-2_2

to university settings and narrowly defined as related to the teaching of Computer Science. Over the last ten to fifteen years, the field has broadened to include many fields and levels of education. The shift from the term *computer science education research* to *computing education research* also reflects the change of scope.

It seems clear from the development of the computing field, and the consequent evolution in computing education over the past 30 years that research plays an essential role in the development of relevant and innovative delivery of computing education. Researchers must be computing professionals with skills in other relevant disciplines capable of identifying fundamental teaching and learning issues unique to computing education. The researchers should be able to focus on meaningful research on those issues and challenges and interpret and disseminate the results. We posit that the key contribution of computing education researchers is their ability to illuminate and address domain-specific teaching and learning issues in computing education. CER researchers are central to identifying relevant research, either in higher education or in computing itself, and applying this to computing education, developing practical insight and new approaches to teaching and learning tailored to the computing education domain.

Given the rapid growth of Computing as a discipline and the complexity of the research foci aligned with educational transformation, it is clear that a single definition of CER is not possible. However, taking a historical perspective, including the development of a sense of scholarship, allows us to analyze the focus of CER over time. Furthermore, we will provide an environmental structure for CER, that includes the components *computing in general*, *learning and teaching computing*, and *educational research*, to discuss the interaction and overlap between CER and the other aspects of the field of Computing. The concept of scholarship gives a common ground for valuing CER. To that end, we provide a short introduction to scholarship based on a framework developed by Glassick [19] as a basis for the CER community.

Finally, we will reflect on the status of CER as a discipline. In this, we will use some criteria for a discipline, presented originally for Science Education Research by Fensham [15] and provide our assessment of how well CER fulfills these criteria. We argue that CER has matured to be seen as a legitimate research discipline, and conclude by relating CER to other examples of Discipline Based Education Research (DBER).

2 The History of CER

As we approach tracing the history of what today is called Computing Education Research (CER), it seems relevant to provide an overview of the organizations and activities that contributed to establishing the field. In particular, the publication venues that developed from the late 1960s to the present and the types of publications and discussions that have been dominant in different phases of development reflect the field's maturation. Other chapters in this book present a more extensive analysis of the publication trends, topic shifts, and influential research groups and authors.

2.1 Dissemination of Results

The Association for Computing Machinery (ACM) Special Interest Group in Computer Science Education (SIGCSE) emerged as an organizational structure within the ACM in the late 1960s, with the first issue of the quarterly newsletter, the SIGCSE Bulletin, appearing in 1969. The first SIGCSE Technical Symposium followed this initiative in November of 1970. There ensued a considerable delay in terms of the internationalization of the SIGCSE conferences and workshops. Scholars outside North America would have to wait until the 1990s and the establishment of the Innovation and Technology in Computer Science Education (ITiCSE) conference series and the Australasian Computing Education Conference (ACE) in 1996, followed by Koli Calling in 2001 and the International Workshop on Computing Education Research (ICER) in 2005. The most recent new SIGCSE-sponsored conference targeting fields outside the US, Europe, and Austral-Asia is CompEd, first organized in 2019. In recent 10 years, also several other new CER-focused conferences have been launched in Europe, such as UKICER and CSERC, as well as WiPCSE which focuses on K-12 computing education.

On the other hand, while the conferences mentioned above have focused solely on CER, they are certainly not the only venues in which CER papers are published. There is a long history of research on professional programming, beginning, for instance, with Weinberg's studies on the psychology of programming [71] and Soloway's empirical studies of programming, e.g. [63], in the 1970s and 1980s. This work has also addressed learning programming, for example, by comparing experts' and novices' conceptions of programming and working patterns. Psychology of programming interest group (PPIG) has organized relevant workshops since 1986, where CER papers are published. Moreover, engineering education conferences, such as Frontiers in Education (FIE), founded in 1971, frequently publish computing education research at the tertiary and school levels. There are several relevant journals that have emerged over time in addition to the conferences. Some of the more influential in the development of the field are the Taylor and Francis publication "Computer Science Education", ACM Transactions on Computing Education (TOCE), and the IEEE Transactions on Education. In addition, numerous educational journals and engineering education journals have accepted CER papers for a long time.

While these venues have mainly focused on adult learning of computing, it is essential to recall early studies on children and programming. Examples are the LOGO language for teaching programming to children developed around 1970 and Papert's seminal work "Mindstorms" in 1980 [46, 62].

In this chapter, we discuss CER from the perspective of its main publishing venues and thematic research foci. For instance, in another chapter in this book, we cover computing in schools.

2.2 Views on CER

In the early years, there was little if any discussion regarding CER and the nature of the field. For instance, general discussions of CER's status as a discipline emerged after the turn of the century. Early studies and research relevant to computing education were not considered a separate research field, CER. In many cases, they were reports that described teaching practice and learning approaches broadly assessed as successful. The discussion of theory, structure, and research expectations first emerged in panel discussions on what CER should be in the early 2000s, followed by the release of two seminal handbooks on CER.

2.2.1 Early Panel Debates

During the early 2000s, there was considerable debate around the nature of computing education research, its structure, and its rigor. An example from 2002 is the description of Computing Education Research (CER) and the models and methods associated with such research [48]. Early discussions of the nature of what has become CER can also be found in the ACM community proceedings of the 2004 SIGCSE symposium and ITiCSE and ACE conferences somewhat later [4, 11, 20, 47].

Two panel debates at ITiCSE 2004 presented perspectives on research. Ben-Ari et al. [4] argued that theoretically sound research enabled generalization of results beyond a single context and allowed comparison and contrasting results from comparable studies in multiple settings. They exemplified with the swathe of multi-national and multi-institutional studies produced from the Bootstrapping project [18, 52]. They also traced the discourse on research methods and rigor back to their work in 1998 [12].

Goldweber et al. [21], on the other hand, presented four perspectives on CS Education research as a field and what it meant to do good research. Martyn Clarke argued that rigorous research could only be conducted with acceptable quality using research methods and interpretation as a collaborative interdisciplinary exercise between educational science and computing. Sally Fincher advanced the view that the field was concerned with noticing phenomena in the computing education context and, through research, providing insight and explanations for both the phenomenon and its impact on learning. Michael Goldweber described the field as about exchanging ideas and innovations among professionals. Arnold Pears summed up the panel by arguing that all of these approaches had merit and that the vital aspect of working in the field was to be honest about the nature of the contribution in relation to the three paradigms presented by the other speakers. Each type of contribution has value to a particular audience. However, they are focused on different aspects of the domain and vary in their level of relevance to other educators.

The main argument was that computing domain expertise was vital to success and that it is essential to devote time to establishing a common understanding of the role of the SIGCSE and ITiCSE communities in defining the scope and focus of CER.

2.2.2 Handbooks on CER

The first book in the field of CER, by Fincher and Petre [16], emerged at about the same time as the above-mentioned panel debates. While they did not give a comprehensive definition of the field, they described the scope and diversity of the work by dividing the field into ten subfields of research. These included *student understanding* which explores students' mental and conceptual models, their perceptions, and misconceptions; *animation, visualization, and simulation* focuses on building and researching various software tools to support teaching and learning various topics in computer science; *teaching methods*, a broad field that covers aspects such as scaffolding learning, supporting interaction, collaboration, and teamwork among students, and different aspects of project work; *assessment* that covers assessment methods, including validity, as well as automated assessment and feedback, and plagiarism detection; *educational technology* focuses on how new learning and teaching platforms and technologies can be used and integrated into computing education, as well as development and impact of tailored learning environments for computing, such as specialized programming environments; *transferring professional practice into the classroom* aims at understanding professional work practices and how these could be applied in computing education; *incorporating new developments and new technologies* explores how recent new developments and technologies in computing can be incorporated into classroom and courses; *transferring from campus-based teaching to distance education* investigates how educational content and methods can be transferred into online settings; *recruitment and retention* is a subfield that explores ways to attract and retain students into computing, including broadening participation by attracting underrepresented minorities and studying equity and diversity issues in computing education contexts; and finally *construction of the discipline*, a subfield that addresses the curriculum development, building academic qualifications, and explores the nature of discipline itself.

Fifteen years later, the Cambridge Handbook of Computing Education Research [17], currently the most comprehensive source for work in CER, splits work in a new way reflecting the development of the field. The first field is *systemic issues*, topics that persist in the field. These include research on introductory and more advanced programming, various pedagogical approaches, assessment and plagiarism research, and questions addressing equity and diversity. The second broad field, labeled as *new milieux*, addresses more recent issues which have arisen when computing education has spread beyond the "traditional" university settings in formal classrooms and departments of computing. The *new milleux* covers research on computational thinking and computing in schools (K-12), computing for other disciplines, and new programming paradigms. A third field is *systems software and technology*, where research focuses on how software and hardware tools, tangible computing, and integrated learning environments can support learning. Finally, *teacher and student knowledge* is an field investigating issues concerning the production and acquisition of computing knowledge, for example, teacher knowledge and teacher training, professional development, learning outside classrooms, student knowledge and misconceptions, students' motivation, attitudes, and dispositions, as well as students as teachers and communicators.

While the CER handbook covers the field widely, there are some additional fields that are not minor and thus worth adding. *Broadening participation* is closely related to equity and diversity, but it also covers work that seeks to find ways to attract new people, especially children, to consider computing as their future choice of study. *Games and gamification* have been widely used in computing education to support students' motivation and encourage studying. *Construction of the discipline* addresses curriculum development, building academic qualifications, and exploring the nature of the CER itself. Finally, *learning analytics and educational data mining* are increasingly used to investigate various aspects of students' behavior while studying by exploring data sets collected with software (e.g., [22, 25]).

The above seminal books split the field into subfields based on the topic of research. Other approaches describing the field have considered the nature or type of work that is carried out, as well as the impact of the work.

2.3 Classification of CER Papers

In 2004, David Valentine published his paper presenting a meta-analysis of 20 years of CS1 papers in SIGCSE proceedings [69]. Valentine produced what can be considered the first attempt to provide a taxonomy of work in CER. He divided the contributions published to date in the SIGCSE proceedings into the following categories: *Experimental* analysis with a focus on positivist experimentation, *Marco Polo* "I went there and saw this" experience reports, *Philosophy* debating a general issue in the field, *Tools* technology in education, *Nifty* cool tips and tricks, and *John Henry* extreme experimentation.

Pears et al. [49] sought to identify the core literature for the field that all CER researchers should be aware of. While this may have been a feasible goal in 2005, the growth and development of the field challenge whether such a goal is relevant anymore. However, in many subfields, it is possible to identify *influential* and *seminal* papers which have guided future work by opening new avenues for research and/or having a clear impact on future work. During the last 10 years, the field has also seen a significant number of *synthesis papers* that summarize and classify substantial work. Most of these are reviews, often systematic reviews, while there are also papers defining taxonomies seeking to categorize work in an field.

Simon [58, 60] developed a different categorization system with four dimensions. He separated the research target into two dimensions. *Context* described the curricular context where the research was carried out (if any), such as programming, software engineering, databases, data structures and algorithms, information systems, or networks. *Theme/Topic*, on the other hand, described what was actually investigated, such as teaching/learning techniques or tools, assessment, students' ability/aptitude, distance/online delivery, ethics/professional issues, gender issues and diversity, recruitment, etc. *Scope* described how wide the community addressed was, ranging from a single course to multi-institutional studies. Finally, the *nature* dimension classified the papers on whether they were collecting and analyzing empirical data, or just reporting on experiences or stating a position.

2.4 Frameworks for CER

Pears proposed that a CER framework was a fundamental first step towards the goal of helping the field to interpret and structure its future. Such a framework would be valuable in the computing education community and in the wider interdisciplinary context of Discipline Based Education Research (DBER) [61], of which CER is a part. This work was further developed through efforts to define the core literature [49] and to analyze and try to establish criteria through which to understand the impact of publications on shaping this emerging field. The fundamental argument behind the core literature effort was establishing an understanding of CER work. Such an understanding was widely agreed to be a vital background for younger colleagues and Ph.D. students in their research. It would help to define what the field was about, and what important publications had shaped the discourse and conduct of research. These research frameworks and a discussion of theory's role in CER are covered in detail in chapter "Theory and Approaches to Computing Education Research".

3 CER Today

Today, CER is an increasingly diverse field. Using Simon's classification scheme [58, 60] as a framework, CER addresses questions relevant in a wide selection of curricular contexts within ACM curriculum recommendations [2] in addition to informal learning of computing. However, the focal fields of research are very much skewed. Introductory programming education is a persistent subfield in CER, which receives a large share of attention. This skew is visible, for example, in the systematic literature review by Luxton-Reilly et al. [30], which identified 1666 papers focusing on this field from 2003 to 2017. Moreover, Robins' review of research on novice programmers [56] in the Cambridge Handbook is the most extensive chapter in the book. In general, many popular topical fields in CER are easy to map to typical first and second-year courses in computing programs. However, in recent years, a fast-growing number of research papers have emerged focusing on K-12 computing education, mostly concerning challenges in learning programming from pre-school to high school levels. CER in schools and the general topical fields in CER are discussed in more depth in two other chapters in this book.

The second dimension, using Simon's classification, is the research theme, i.e., what the research is about in the specific context. This theme includes numerous aspects, such as ability, aptitude, assessment techniques or tools, cheating and plagiarism, curriculum, distance/online delivery, educational technology, ethics/professional issues, gender issues, recruitment, teaching/learning techniques or tools etc. (for a more comprehensive list see appendix in [56]). All of these aspects are addressed in CER.

Thirdly, CER covers studies with very different scopes. A large share of studies focuses naturally on course-level research contexts or even task-level contexts. However, on the other end, highly important international studies have collected and analyzed data from multiple institutes internationally, e.g., [29, 40, 68].

In recent years, CER has increasingly paid attention to research rigor, which essentially concerns applying rigorous data collection and analysis methods and using theoretical frameworks to guide research designs and interpretation of results. In addition, improving the quality of reporting has been addressed in many ways, for example, with explicit definitions of research questions in papers, more accurate reporting of applied methods and theories, and stating limitations more clearly. This increase in rigor is also visible in more specific instructions for authors on conference and journal websites, more explicit review criteria, and improved rigor in review processes [53] as well as in methodological reviews of the CER literature, e.g., [27, 31, 38, 41, 55].

Parallel with this, there has emerged a growing interest in the analysis and discussion of the use of theories to inform CER. Multiple reviews focusing on theories in CER have appeared in recent years [27, 32, 34, 64, 65]. Computer Science Education published a special issue, "Advancing Theory about the Novice Programmer", in 2019 with several papers addressing this topic. ACM Transactions on Computing Education publishes two special issues on "Conceptualizing and Using Theory in Computing Education Research" in 2022–2023 [67]. Moreover, there is an increasing interest in developing domain-specific theories and models in CER that emerge from the field to explain the complex phenomena related to teaching and learning computing [33, 35]. Examples are statistical models, such as structural equation models or regression models, qualitative grounded theories, or phenomenographical outcome spaces. Though, more elaborated theories have emerged, such as a theory of instructing programming skills by Xie et al. [75] or models for predicting student success based on their programming behavior and social aspects [7, 8] by Carter et al. There is also discussion on the role of theories in CER, whether they are always needed and if they even restrict designing new learning innovations [44].

All this demonstrates that the field is maturing as a research field. Rigorous research papers have always been published in the field. However, experience reports or practice papers dominated most CER-focused venues in the first decades. During the last 20 years, the share of research papers among publications has been steadily increasing [59], and a growing awareness and emphasis have emerged on research quality. This book discusses these developments in more detail in another chapter focusing on meta-studies in the field.

4 An Environment for CER

To complement the above, in this section, we present a more generic structure that places CER in an environment by relating CER to aspects that are informing and/or are being informed by CER. The environment in which CER exists has the following components:

- Computing in general;
- Education research in general;
- Learning and teaching computing.

This structure has *Computing in general* as an essential object that informs CER. *Education research*, which includes any relevant research approach, provides CER with many relevant methods and theories, not least related to learning and collaboration. *Learning and teaching computing* is where CER "results" are received and also a study object for CER. Learning and teaching computing is not restricted to higher education. There is, for instance, a large body of work on non-formal learning computing and computational thinking. Discussing the boundaries and overlap between these elements in the proposed structure will provide a way to understand CER.

4.1 CER and Computing in General

Computing is essential to CER as it is the object to be understood in educational contexts. A confounding factor is that computing is a vast field. ACM curricula 2020 identifies seven subfields: computer engineering, computer science, software engineering, information systems, information technology, cybersecurity, and data science [2]. Research that could be included as CER can be carried out in any of these subdomains. Another aspect of computing is that it is used as a component in other, often quite complex, contexts. There is, thus, a need to understand how computing artifacts interact and influence various settings. This blurs the lines between what is computing and what is not. In CER, a wider definition of computing which includes understanding its role in complex contexts, is appropriate.

In the past, CER focused much on teaching and learning computer science topics, especially programming [6] and other basic undergraduate studies, such as data structures, algorithms, and databases. Research on advanced computer science topics has been rare. A fairly obvious reason is that the main challenges in computing courses focus on the basic courses, where many students struggle and drop out. These courses are also often large in the number of enrolled students (ranging to hundreds or even thousands). In contrast, advanced courses have a small or moderate size, which enables more personal level support for students.

However, as computing has widened as a domain field, also CER has diversified. When computing subdisciplines have established their publication venues, it is natural that CER papers addressing those subdisciplines are published in these conferences and workshops. An example is software engineering education research published in the educational track of ICSE (International Conference on Software Engineering). Similarly, programming education research papers in engineering education contexts are often published in engineering education venues. This diversity forms a challenge in defining the field because the label CER is often used in a narrow perspective which focuses on "CER only" publication venues.

If we contrast CER and computing research, we can identify similarities and major differences worth noting. CER can be characterized as a social science that heavily deals with human participants. As such, it applies many similar methods and theoretical frameworks used in HCI research, usability studies, and empirical software engineering. On the other hand, a sizable subfield of CER is research related to developing advanced software for education, which is close to software engineering research. Some of such work is plain engineering, i.e., designing and implementing new software. At the same time, more specific technical research exists, where novel technological solutions are developed, e.g., for supporting the integration of interactive learning content [5].

Theories have quite different interpretations in CER and computing sciences. In CER, theories focus on describing, explaining and/or predicting human behavior, thus working in the domain of social sciences. On the other hand, theoretical computer science, algorithms research and cryptography strongly focus on theorems and general statements based on the mathematical research tradition. Data science, machine learning, and data mining draw much from Statistics research, thus being also close to mathematics research tradition. In addition, in technical fields of computing, for instance, software technology, computer engineering, and cybersecurity, mathematical and engineering traditions are strong, too.

4.2 CER and Education Research in General

Education together with computing are the "parents" of CER. Education research generally addresses such aspects of teaching and learning which are not tied to some specific discipline. In contrast, Disciplinary Education Research, such as CER, focuses on education-related topics in its specific disciplines. In CER, it can, for instance, be researching learning obstacles or fruitful ways to motivate learners on a particular topic in computing. Other indirect examples are studying the influence of culture and understanding diversity issues in computing education. The aspects of computing can vary, but computing has to be a component for research to be CER, and CER researchers need to understand the computing aspect.

CER borrows and applies much work from Education, Psychology, and other social sciences like Sociology. Numerous theoretical frameworks have been adopted from these social sciences to support the investigation of various phenomena in computing education [32, 65]. Moreover, CER applies many research methods, e.g., qualitative methods such as grounded theory, phenomenography, phenomenology, and various types of content analysis, that have been used in the social sciences for a long time.

The question can be raised, why is CER not just a subfield of Education? There are several arguments against this claim. First, CER addresses computing-specific concepts, processes, and phenomena, and their investigation requires a deep understanding of these topics. Without a proper understanding of concepts such as objects, classes, recursion, and notional machines, it is impossible to study how students understand the topics. On the other hand, there is certainly work in CER, where this requirement is less important, such as investigating students' team working in capstone projects or computing students' experiences in company internships. However, it is natural that such research focusing on computing education is published in related venues.

CER also increasingly develops its theoretical frameworks in terms of various statistical models of data collected from computing education settings or qualitative analyses of such data [33, 34]. Much work is devoted to building new validated instruments or concept inventories that address learning computing topics or adapting existing more general instruments from Education or Psychology [34] into computing contexts.

Thus, CER is gradually building its own set of theoretical tools and methods strictly tied to knowledge and skills in computing, which general theories and instruments cannot reach.

4.3 CER and Learning and Teaching Computing

Learning and teaching computing is in some way or another the ultimate target of CER as it is its field of practice. This field of practice is even more complex than the computing concept. Part of the complexity stems from the complexity of computing itself, but further complexity arises from aspects such as the influence of educational contexts, formal vs. non-formal education, specialized vs. integrated learning objectives, and higher vs. K-12 education. The boundary, or rather overlap, between CER and its field of practice is unclear. For instance, many CER researchers are expected to provide concrete help to computing teachers.

4.3.1 Research vs Development

There is sometimes a distinction between development and research made, especially concerning funding opportunities. This line is problematic in many cases since many CER projects draw on action research where development is part of the research. Furthermore, the difference between computing education research and development of computing education is subtle since computing teachers carry out development in more or less scholarly ways. Scholarship is a concept that addresses the identity of teachers and provides a common ground to understand how development and research are related and complementary with regard to influencing teachers.

Glassick's definition of scholarship provides a basis to calibrate our view of CER and the maturity of the field [19]. He discusses the standards to which responsible scholarship in a discipline might be held and establishes a framework for scholarly quality in six fields. In his view, high-value scholarship should excel in all six fields.

- Clear Goals—Does the scholar clearly state the basic purposes of his or her work? Does the scholar define objectives that are realistic and achievable?
- Adequate Preparation—Does the scholar show an understanding of existing scholarship in the field? Does the scholar bring the necessary skills to their work?
- Appropriate Methods—Does the scholar use methods appropriate to the goals? Does the scholar effectively apply the methods selected?
- Significant Results—Does the scholar's work add consequentially to the field? Does the scholar's work open additional fields for further exploration?
- Effective Presentation—Does the scholar use appropriate forums for communicating work to its intended audiences? Does the scholar present her message with clarity and integrity?
- Reflective Critique—Does the scholar critically evaluate his or her work? Does the scholar use evaluation to improve the quality of future work?

This framework provides a basis to reflect upon CER publications and the nature of the field's strategic research discourse.

4.3.2 Research Publication or Not?

There is a long tradition of publishing papers that present novel pedagogical innovations, experiences of applying some existing pedagogical approaches and methods, novel learning resources, and software tools to support education. Many of these include not only a description of these but also some form of evaluation, e.g., student feedback, course or task results, or teacher's reflections. There may be a pre/post-test analysis of learning gains in a group or a comparison with previous years' cohort results. There is no consensus in the field which publications should be considered *research papers* that would be counted as CER contributions, and which are *practice reports, experience reports, Marco Polo papers* or some other names to identify them as something different.

This split causes tensions within the community, as it suggests that some published papers may be of poor quality. Yet, they may be highly valuable for teachers looking for new ideas and resources for their teaching. However, there is a clear trend in the field that the share of research papers is increasing in the main publishing venues. See, for example, [59] and chapter "The Evolution of Computing Education Research: A Meta-Analytic Perspective" in this book. This change is also reflected in the instructions for authors in journals and conference calls for papers. The expectations for publications have become more specific, typically emphasizing the use of theoretical frameworks to support arguments and analyses, as well as applying rigorous empirical methodologies and styles of reporting results. Many conferences have developed their review processes to better address such requirements while there is also debate on what is the main role and purpose of conferences—presenting research publications, or meeting people, and sharing ideas and innovations [53] as well as what is the role of theory in CER [44]. A part of this debate follows from the practice in computing sciences, where conference papers are considered valuable scientific merit, sometimes more valuable than journal papers. In almost all other disciplines,research contributions are mainly published in journals, and conferences are venues for presenting novel work and meeting people.

There is no sign that this tension would be vanishing, and yet both types of contributions are needed to make progress in improving computing education.

4.3.3 CER and Educational Settings

The environment in which learning and/or teaching computing occurs has evolved from being done exclusively at higher education institutions in the early days. Today, aspects of computing are integrated in the school system and are part of most disciplines and degree programs at the higher education level. There is also substantial work on understanding educational settings other than the formal settings, for instance, coding clubs and lifelong learning.

Learning Outside Formal Settings While much research has focused on the school environment and formal education, there is also considerable work done in the informal arena. In the last decade, there has been considerable research effort placed into the impact of societal initiatives, e.g. the MicroBit and Arduino systems, learning in informal settings, the Maker movement, Code Dojos, Bebras, Hour of Code, CS for All, and CS Unplugged to name just a few. Informal learning in computing has also been the focus of considerable research, including the EU-funded ComNPlay Science project. Publication of non-formal, informal, and school-level CER has often been in different venues than those utilized by tertiary CER researchers. Venues of interest to these communities include a range of school-related conferences in Europe, Asia, and the Americas, often published in languages other than English. These activities and related research have increasingly found their way into the last decade's traditional ACM and IEEE publication venues.

Learning in Integrated Settings Early work by Papert in 1980 on Computational Thinking (CT) was reawakened by Jeanette Wing in 2006 [74], who emphasized the breadth of contexts where CT is needed. The CT trend has spawned a plethora of works discussing the relevance and definition of CT [13, 51] and what that might entail [37, 45, 50]. CT is also linked to informatics [9] and other European disciplinary fields related to computation and computing machinery, as well as the teaching of these concepts in the various levels of compulsory schooling [10, 36].

CT is also connected to future computing and a broader computing milieu, embracing concepts like virtual and augmented reality, artificial intelligence, and biotechnologies. This increase in technology adoption means that future citizens should be familiar with the above mentioned concepts to stay competitive in the job market.

In addressing these needs, educational institutions have an important role in preparing future entrepreneurs through appropriate curricula and methods that appeal to today's learners. The 2021 World Economic Forum report, a Swedish National Report on Digital Cutting Edge Competence [24, 73], and an influential study by Tedre et al. [66] show that there is a need for a nuanced understanding of CT, calling for extension and contextualization of CT to include explicitly Machine Learning (ML) and AI (CT 2.0). Munasinghe [43] has recently shown that many concepts related to CT should be further elucidated to facilitate their accessibility to teachers. Thus, computational thinking has been actively promoted in schools in an integrative approach, helping to enhance our definition of STEAM (Science, Technology, Engineering, the Arts, and Mathematics) education. STEAM education provides several benefits, such as enhancing the skills of analysis and problem-solving and creativity enhancement [3, 70].

However, there are still some challenges and issues in STEAM-related activities integration into school and within STEAM subjects. Researchers and educators are consequently re-examining the importance of STEAM-related activities and programs, specifically developing computational thinking skills and integrating maker education where learners imagine, design, and create projects that combine learning content with practical hands-on applications. Our approach to STEAM also emphasizes collaboration and integration. These frameworks draw on the work of Yang et al. [76], particularly in terms of how CT could be positioned in the curriculum by designing and implementing an integrated STEM and CT lesson. Yang emphasizes the role of a problem-based process for integrating CT problems and solutions into after-school programs using hands-on inquiry activities which are exciting and engaging for students. Moreover, programming and using physical computing objects—robots enable students to engage in scientific practices and, in such a way, learn some engineering aspects (such as bridge design) and gain satisfying experiences.

The design process takes place through *design thinking* (DT), which realizes learning through experience and complex problem solving for motivation, openness to new ideas, and creative thinking in the learning process [57]. Thus, in the literature, the need to adapt the design to learning is often emphasized by scholars and practitioners [26]. Design thinking is seen in this context as a learning design that facilitates a constructive way of learning due to practice-based development of a range of target skills [57]. Learners receive several benefits from integrating STEAM learning with computing by modeling various phenomena, such as computational thinking learning [23]. However, researchers and educators still face the challenge of defining computational thinking and getting a theoretical grounding for what form it should take in school. One of the possible solutions proposed by Weintrop and colleagues is to develop CT taxonomy [72]. In such a way, CT can be embedded in the STEAM subjects' context. CT development is then made available through STEAM-related activities integration in school, especially through hands-on projects [39].

5 Is CER a Discipline?

Major academic disciplines, such as history, linguistics, mathematics, and computing, comprise further lower-level disciplines. It is easy to think that each lower-level discipline emerged and was accepted. In computing, for example, few would doubt that cybersecurity and data science are legitimate research disciplines, although they have emerged recently. Why is there a need to question whether computing education research is a research discipline?

One reason is that many academics lead what Lister [28] has called a 'double life': they educate students, conduct research, and keep these two activities well apart. It simply does not occur to these academics that work regarding one's teaching, and one's students' learning might be researching.

This 'double life' need not matter, except that it affects such matters as research funding, research-teaching load balance, and promotion prospects. These reasons are why computing education researchers feel the need to persuade other computing researchers that what they are doing is indeed research. One way to do this is to establish that computing education is a legitimate research discipline.

Computing is not the only field whose educational researchers have felt this need. In 2004, Fensham [15] set out the criteria by which a DBER might be assessed to determine whether it is a research discipline. We very briefly review these criteria here.

Structural criteria focus on organizational things.

- There are many examples of academic recognition regarding professorships in computing education research or very close titles for the target field.
- Two prestigious research journals, ACM Transactions on Computing Education and Computer Science Education, are focused on the field, and numerous journals in closely related fields regularly publish CER papers.
- There are professional associations in the field, such as SIGCSE.

- Several research conferences, such as ICER and Koli Calling, focus solely on CER.
- There are research centers focusing on CER, for example the Center for Computing Education Research at the IT-University of Copenhagen [1]. There are also numerous longtime research groups globally that focus all or most of their work on CER.
- There are regular research training activities, especially doctoral consortia associated with major conferences in the field.

Intra-research criteria focus on how the actual research is carried out.

- There is a substantial amount of scientific knowledge in the field, evidenced by the huge number of papers in the data pool. Applying this information for designing and implementing new research requires substantial expertise in the field.
- CER asks questions that can be answered only in the CER domain, e.g., how students learn programming and what misconceptions they might have.
- CER carries out its own conceptual and theoretical development, as demonstrated, e.g., in [33, 34].
- The field develops its own research methodologies, for example, different types of log data collection and analysis of programming process data, creates its own concept analysis instruments [54], and adapts methods and instruments from other disciplines, e.g. [14, 42].
- The field has much work which builds on previous work, thus demonstrating progression.
- CER has many highly cited model publications guiding future research and seminal publications opening new insights, as mentioned above in [49].

In his doctoral thesis [59], Simon examined Fensham's criteria in detail, and his findings are summarized briefly in Sect. 2 of chapter "The Evolution of Computing Education Research: A Meta-Analytic Perspective" of this book. In short, the conclusion is that according to Fensham's criteria, CER is indeed a legitimate research discipline.

6 CER and Other DBER Disciplines

Computing degree programs often include studies of other disciplines that support learning computing, e.g., studies in some fields of mathematics and logic. Moreover, computing students often study other fields as their minor or voluntary studies, e.g., physics, economics, languages, etc. Therefore, it is natural that teaching and learning some topics in these fields can also be addressed in CER contexts when the target group is computing students. Correspondingly, many students from other fields take computing, especially programming courses, and include them in their degrees. Depending on the institutions, such courses can be given by computing teachers in a department of computer science (or similar), as service courses or minor studies, or the courses can be organized independently by their departments, e.g., in engineering schools or departments.

This overlap possibly occurs most frequently in engineering studies. Therefore, there is a clear overlap between computing education research and Engineering Education Research (EER). CER papers are frequently published in EER conferences and journals, which is natural.

It is difficult to state clear differences between CER, EER, and other DBER disciplines except in the main domain of investigation. Similar research methods and the same theoretical frameworks from social sciences can be used in all of them.

7 Conclusions

CER is an inherently interdisciplinary research field combining research findings, theories and research methods from several other fields in addition to developing its own ones. Therefore, the question of what CER is, where it belongs, and even if it is an independent research field is complex. There is no definite answer to this question but rather a set of complementary and partly contradictory answers. The answers are derived from different agendas regarding creating legitimacy for the field, not least economical in relation to funding bodies. This is an interesting question that is interesting to study further in future work.

From a high-level perspective, CER is one research field in a family of Discipline Based Education Research (DBER) fields. The DBERs are based on education research and informed by their respective discipline. They also rely on theories and methods from relevant other disciplines, such as learning sciences, psychology, and sociology. However, the nature and scope of CER are not so well defined when looking more closely at the field. As discussed above in Sect. 2, the panel discussions in ITiCSE 2004 already revealed many different perspectives on defining the field.

In their seminal book, Fincher and Petre [16] discussed the nature of work in the field using a derivation of Pasteur's quadrant as their framework. They split the publication space into four fields with two axes, one addressing how much the paper was based on evidence, especially empirical evidence, and the other one, how much it was based on argumentation or theory. Thus, papers with high argumentation/theory and low evidence were considered *perspective pieces*, while papers with strong evidence but low in theory were considered *practice papers*. CER papers should focus on the quadrant with strong evidence and strong argumentation/theory. Followed by this, they presented the ten subfields of CER, as described in Sect. 2.2.2. Moreover, in the first part of their book, they discussed how to design and carry out research that would match the requirements in the CER quadrant. Their definition thus focuses on describing two relevant aspects of the field: quality of research and what are the research topics in the field.

The Cambridge Handbook of CER [17] provides a similar, though more descriptive than definitive answer to the question of what CER is. In chapter "What is

Computing Education Research (CER)" of the handbook, "Computing Education Research Today", past and present Editors-in Chief of two prominent journals, Computer Science Education and ACM Transactions on Computing Education, discuss the state of the field. Almost the whole discussion focuses on quality aspects of submitted papers, also considering the historical development of the field. The scope of the field—what is included in CER—is not discussed, while the notion that some submissions are "out of scope" is mentioned. Followed by this chapter, the Handbook elaborates on in-depth research designs, methods, and theoretical frameworks, i.e., research quality aspects, followed by presenting a wide coverage of research topics in the field.

One might even argue whether the goal of defining the field is meaningful at all, as it evolves steadily. Furthermore, the domain field Computing is not a static field but evolves and widens rapidly; thus, Computing Education and CER necessarily follow this development. Pragmatic approaches (such as Fincher and Petre's book and the Cambridge Handbook) which focus on the quality of research work and describe the most relevant current research topics, are actually quite suitable for capturing a "snapshot" of CER. That said, we do believe there is a place for future work regarding classification and definitions of the field, not least as it can provide support for future CER researchers.

One example of how the pragmatic approach can be condensed into a few sentences is how Uppsala Computing Education Research Group [UpCERG, see chapter "A Case Study: The Uppsala Computing Education Research Group (UpCERG)" in this book for more on this group] in the field presents its field of work.

CER addresses learning and teaching in the computing discipline. It is founded on an understanding of the discipline using theories and methods from education research and other relevant disciplines, such as psychology and sociology. Typical fields of study (related to the computing discipline) are learning and teaching core concepts and skills, curricula development, intercultural and interdisciplinary collaboration, identities, and inclusion. The educational context has a focus on higher education, but K-12 and lifelong learning, as well as both formal and informal learning, are also addressed.

References

1. Center for Computing Education Research at the IT-University of Copenhagen. https://ccer.itu.dk/. Accessed: 2022-09-14
2. ACM/IEEE Taskforce: ACM curricula recommendations. https://www.acm.org/education/curricula-recommendations. Accessed: 2022-09-14
3. Aithal, P.S., Aithal, S.: Innovation in B.Tech. Curriculum as B.Tech. (Hons) by integrating STEAM, ESEP & IPR features. International Journal of Case Studies in Business, IT, and Education (2019). DOI https://doi.org/10.2139/ssrn.3406824. URL https://papers.ssrn.com/abstract=3406824
4. Ben-Ari, M., Berglund, A., Booth, S., Holmboe, C.: What Do We Mean by Theoretically Sound Research in Computer Science Education? In: Proceedings of the 9th Annual SIGCSE

Conference on Innovation and Technology in Computer Science Education, ITiCSE '04, pp. 230–231. Association for Computing Machinery, New York, NY, USA (2004). URL https://doi.org/10.1145/1007996.1008059. Event-place: Leeds, United Kingdom
5. Brusilovsky, P., Edwards, S., Kumar, A., Malmi, L., Benotti, L., Buck, D., Ihantola, P., Prince, R., Sirkiä, T., Sosnovsky, S., et al.: Increasing adoption of smart learning content for computer science education. In: Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference, pp. 31–57 (2014)
6. Carbone, A., de Raadt, M., Lister, R., Hamilton, M., Sheard, J.: Classifying computing education papers: process and results. In: Proceedings of the Fourth International Workshop on Computing Education Research, ICER, pp. 161–172 (2008)
7. Carter, A.S., Hundhausen, C.D., Adesope, O.: The normalized programming state model: predicting student performance in computing courses based on programming behavior. In: 11th International Computing Education Research Conference, ICER 2015, pp. 141–150 (2015). URL http://doi.acm.org/10.1145/2787622.2787710
8. Carter, A.S., Hundhausen, C.D., Adesope, O.: Blending measures of programming and social behavior into predictive models of student achievement in early computing courses. ACM Transactions on Computing Education (TOCE) **17**(3), 12 (2017)
9. Dagiene, V., Stupuriene, G.: Informatics Concepts and Computational Thinking in K-12 Education: A Lithuanian Perspective. Journal of Information Processing **24**(4), 732–739 (2016)
10. Dagienė Valentina, Jevsikova Tatjana, Stupurienė Gabrielė, Juškevičienė Anita: Teaching computational thinking in primary schools: Worldwide trends and teachers' attitudes. Computer Science and Information Systems **19**(1), 1–24 (2022).
11. Daniels, M., Pears, A.: Models and methods for computing education research. Australian Computer Science Communications **34**(2), 95–102 (2012)
12. Daniels, M., Petre, M., Berglund, A.: Building a Rigorous Research Agenda into Changes to Teaching. In: Proceedings of Third Australasian Computer Science Education Conference ACE (1998)
13. Denning, P.J., Tedre, M.: Computational Thinking: A Disciplinary Perspective. Informatics in Education (2021). DOI https://doi.org/10.15388/infedu.2021.21. URL https://infedu.vu.lt/journal/INFEDU/article/701. Publisher: Vilnius University Institute of Data Science and Digital Technologies
14. Dorn, B., Elliott Tew, A.: Empirical validation and application of the computing attitudes survey. Computer Science Education **25**(1), 1–36 (2015)
15. Fensham, P.J.: Defining an Identity: The Evolution of Science Education as a Field of Research. Springer Science & Business Media (2004)
16. Fincher, S., Petre, M.: Computer Science Education Research. Routledge Falmer (2004). URL http://www.cs.kent.ac.uk/pubs/2004/1819
17. Fincher, S.A., Robins, A.V.: The Cambridge handbook of computing education research. Cambridge University Press (2019)
18. Fincher, S., Tenenberg, J.: Using Theory to Inform Capacity-Building: Bootstrapping Communities of Practice in Computer Science Education Research. Journal of Engineering Education **95**(4), 265–277 (2006). DOI https://doi.org/10.1002/j.2168-9830.2006.tb00902.x
19. Glassick, C.E., Huber, M.T., Maeroff, G.I., Boyer, E.L.: Scholarship assessed: evaluation of the professoriate. Jossey-Bass, San Francisco (1997)
20. Goldweber, M., Clark, M., Fincher, S., Pears, A.: The relationship between CS education research and the SIGCSE community. In: ITiCSE '04: Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, pp. 228–229. ACM Press, Leeds, United Kingdom (2004). DOI http://doi.acm.org/10.1145/1007996.1008057
21. Goldweber, M., Clark, M., Fincher, S., Pears, A.: The Relationship between CS Education Research and the SIGCSE Community. In: Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education, ITiCSE '04, pp. 228–229. Association for Computing Machinery, New York, NY, USA (2004). URL https://doi.org/10.1145/1007996.1008057. Event-place: Leeds, United Kingdom

22. Grover, S., Korhonen, A.: Unlocking the potential of learning analytics in computing education. ACM Transactions on Computing Education (TOCE) **17**(3), 1–4 (2017)
23. Grover, S., Fisler, K., Lee, I., Yadav, A.: Integrating Computing and Computational Thinking into K-12 STEM Learning. In: Proceedings of the 51st ACM Technical Symposium on Computer Science Education, pp. 481–482. Association for Computing Machinery, New York, NY, USA (2020). URL https://doi.org/10.1145/3328778.3366970
24. Gulliksen, J., Cajander, Å., Pears, A., Wiggberg, M.: Digital spetskompetens – den nya renässansmänniskan: Genomlysning, definition, prognosverktyg och rekommendationer för framtida utveckling. ISBN: 978-91-88961-58-7. Tillväxtverket (2020). Publication Title: https://digitalspetskompetens.se/wp-content/uploads/2020/06/DigitalSpetskompetens_Definition_Gulliksenetal.pdf
25. Hellas, A., Ihantola, P., Petersen, A., Ajanovski, V.V., Gutica, M., Hynninen, T., Knutas, A., Leinonen, J., Messom, C., Liao, S.N.: Predicting academic performance: a systematic literature review. In: Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education, pp. 175–199 (2018)
26. Kirschner, P.A.: Do we need teachers as designers of technology enhanced learning? Instructional Science **43**(2), 309–322 (2015). URL https://doi.org/10.1007/s11251-015-9346-9
27. Lishinski, A., Good, J., Sands, P., Yadav, A.: Methodological rigor and theoretical foundations of CS education research. In: Proceedings of the 2016 ACM conference on international computing education research, pp. 161–169 (2016)
28. Lister, R.: After the gold rush: Toward sustainable scholarship in computing (keynote address). In: Tenth Australasian Computing Education Conference, ACE 2008, pp. 3–17 (2008)
29. Lister, R., Adams, E.S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J.E., Sanders, K., Seppälä, O., Simon, B., Thomas, L.: A multi-national study of reading and tracing skills in novice programmers. In: ITiCSE-WGR ’04: Working group reports from ITiCSE on Innovation and technology in computer science education, pp. 119–150. ACM, Leeds, United Kingdom (2004). DOI http://doi.acm.org/10.1145/1044550.1041673
30. Luxton-Reilly, A., Albluwi, I., Becker, B.A., Giannakos, M., Kumar, A.N., Ott, L., Paterson, J., Scott, M.J., Sheard, J., Szabo, C.: Introductory programming: a systematic literature review. In: Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, pp. 55–106 (2018)
31. Malmi, L., Sheard, J., Bednarik, R., Helminen, J., Korhonen, A., Myller, N., Sorva, J., Taherkhani, A.: Characterizing research in computing education: a preliminary analysis of the literature. In: Proceedings of the Sixth international workshop on Computing education research, pp. 3–12 (2010)
32. Malmi, L., Sheard, J., Simon, Bednarik, R., Helminen, J., Kinnunen, P., Korhonen, A., Myller, N., Sorva, J., Taherkhani, A.: Theoretical underpinnings of computing education research: what is the evidence? In: Tenth International Computing Education Research Conference, ICER 2014, pp. 27–34 (2014)
33. Malmi, L., Sheard, J., Kinnunen, P., Simon, Sinclair, J.: Computing education theories: what are they and how are they used? In: 15th International Computing Education Research Conference, ICER 2019, pp. 187–197 (2019)
34. Malmi, L., Sheard, J., Kinnunen, P., Simon, Sinclair, J.: Theories and models of emotions, attitudes, and self-efficacy in the context of programming education. In: 16th International Computing Education Research Conference, ICER 2020, p. 36–47 (2020)
35. Malmi, L., Sheard, J., Kinnunen, P., Sinclair, J.: Development and use of domain-specific learning theories, models and instruments in computing education. ACM Transactions on Computing Education (TOCE) (2022)
36. Mannila, L., Dagiene, V., Demo, B., Grgurina, N., Mirolo, C., Rolandsson, L., Settle, A.: Computational Thinking in K-9 Education. In: Proceedings of the Working Group Reports of the 2014 on Innovation & Technology in Computer Science Education Conference, ITiCSE-WGR ’14, pp. 1–29. ACM, Uppsala, Sweden (2014). URL http://doi.acm.org/10.1145/2713609.2713610

37. Mannila, L., Nordén, L.å., Pears, A.: Digital Competence, Teacher Self-Efficacy and Training Needs. In: Proceedings of the 2018 ACM Conference on International Computing Education Research, pp. 78–85. ACM (2018). DOI https://doi.org/10.1145/3230977.3230993
38. Margulieux, L., Ketenci, T.A., Decker, A.: Review of measurements used in computing education research and suggestions for increasing standardization. Computer Science Education **29**(1), 49–78 (2019)
39. Martín-Ramos, P., Lopes, M.J., Lima da Silva, M.M., Gomes, P.E.B., Pereira da Silva, P.S., Domingues, J.P.P., Ramos Silva, M.: First exposure to Arduino through peer-coaching: Impact on students' attitudes towards programming. Computers in Human Behavior **76**, 51–58 (2017). DOI https://doi.org/10.1016/j.chb.2017.07.007. URL https://www.sciencedirect.com/science/article/pii/S0747563217304193
40. McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y.B.D., Laxer, C., Thomas, L., Utting, I., Wilusz, T.: A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. SIGCSE Bulletin **33**(4), 125–180 (2001). DOI http://doi.acm.org/10.1145/572139.572181
41. McGill, M.M., Decker, A.: A gap analysis of statistical data reporting in K-12 computing education research: recommendations for improvement. In: Proceedings of 51st SIGCSE Technical Symposium on Computer Science Education, pp. 591–597 (2020)
42. Morrison, B.B., Dorn, B., Guzdial, M.: Measuring cognitive load in introductory cs: adaptation of an instrument. In: Proceedings of the tenth annual conference on International computing education research, pp. 131–138 (2014)
43. Munasinghe, B., Bell, T., Robins, A.: Teachers' understanding of technical terms in a Computational Thinking curriculum. In: Australasian Computing Education Conference, ACE '21, pp. 106–114. Association for Computing Machinery, New York, NY, USA (2021). URL https://doi.org/10.1145/3441636.3442311
44. Nelson, G.L., Ko, A.J.: On use of theory in computing education research. In: Proceedings of the 2018 ACM Conference on International Computing Education Research, pp. 31–39 (2018)
45. Niemelä, P., Pears, A., Dagienė, V., Laanpere, M.: Computational Thinking – Forces Shaping Curriculum and Policy in Finland, Sweden and the Baltic Countries. In: D. Passey, D. Leahy, L. Williams, J. Holvikivi, M. Ruohonen (eds.) Digital Transformation of Education and Learning - Past, Present and Future, IFIP Advances in Information and Communication Technology, pp. 131–143. Springer International Publishing, Cham (2022). DOI https://doi.org/10.1007/978-3-030-97986-7_11
46. Papert, S.: Mindstorms: Computers, children, and powerful ideas. NY: Basic Books (1980)
47. Pears, A., Daniels, M.: Developing Global Teamwork Skills: The Runestone Project. In: M. Castro, E. Tovar, M.E. Auer (eds.) IEEE EDUCON 2010 – The Future of Global Learning in Engineering Education (2010)
48. Pears, A., Daniels, M., Berglund: Describing Computer Science Education Research: An Academic Process View. In: Conference on Simulation and Multimedia in Engineering Education, ICSEE'2002, San Antonio, Texas, pp. 99–104. Society for Computer Simulation International (2002). URL PearICSEE2002.pdf
49. Pears, A., Seidman, S., Eney, C., Kinnunen, P., Malmi, L.: Constructing a Core Literature for Computing Education Research. ACM SIGCSE Bulletin **37**(4), 152–161 (2005). DOI https://doi.org/10.1145/1113847.1113893
50. Pears, A., Dagiene, V., Jasute, E.: Baltic and Nordic K-12 Teacher Perspectives on Computational Thinking and Computing. In: V. Dagien\.e, A. Hellas (eds.) Informatics in Schools: Focus on Learning Programming: 10th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2017, Helsinki, Finland, November 13–15, 2017, Proceedings, pp. 141–152. Springer International Publishing, Cham (2017). URL https://doi.org/10.1007/978-3-319-71483-7_12
51. Pears, A., Tedre, M., Valtonen, T., Vartiainen, H.: What Makes Computational Thinking so Troublesome? In: 2021 IEEE Frontiers in Education Conference (FIE), pp. 1–7 (2021). DOI https://doi.org/10.1109/FIE49875.2021.9637416. ISSN: 2377-634X
52. Petre, M., Fincher, S.: Bootstrapping Research in Computer Science Education. In: Tacoma and Port Townsend, Washington, USA (2002). URL http://depts.washington.edu/bootstrp/

53. Petre, M., Sanders, K., McCartney, R., Ahmadzadeh, M., Connolly, C., Hamouda, S., Harrington, B., Lumbroso, J., Maguire, J., Malmi, L., et al.: Mapping the landscape of peer review in computing education research. In: Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education, pp. 173–209. ACM (2020)
54. Porter, L., Zingaro, D., Liao, S.N., Taylor, C., Webb, K.C., Lee, C., Clancy, M.: BDSI: A validated concept inventory for basic data structures. In: Proceedings of the 2019 ACM Conference on International Computing Education Research, pp. 111–119 (2019)
55. Randolph, J.J., Julnes, G., Sutinen, E., Lehman, S.: A methodological review of computer science education research. Journal of Information Technology Education: Research **7**(1), 135–162 (2008)
56. Robins, A.V.: Novice programmers and introductory programming. In: S. Fincher, A. Robins (eds.) The Cambridge handbook of computing education research, pp. 327–377. Cambridge University Press (2019)
57. Scheer, A., Noweski, C., Meinel, C.: Transforming Constructivist Learning into Action: Design Thinking in education. Design and Technology Education: an International Journal **17**(3) (2012). URL https://ojs.lboro.ac.uk/DATE/article/view/1758
58. Simon: A Classification of Recent Australasian Computing Education Publications. Computer Science Education **17**(3), 155 – 169 (2007). URL http://www.informaworld.com/10.1080/08993400701538021
59. Simon: Emergence of computing education as a research discipline. Ph.D. thesis, Aalto Univerity, Finland (2015)
60. Simon, Carbone, A., Raadt, M.d., Lister, R., Hamilton, M., Sheard, J.: Classifying Computing Education Papers: Process and Results. In: R. Lister, M. Caspersen, M. Clancy (eds.) Fourth International Computing Education Research Workshop (ICER 2008). ACM Press, Sydney, Australia (2008)
61. Singer, S.R., Nielsen, N.R., Schweingruber, H.A.: Discipline-Based Education Research: Understanding and Improving Learning in Undergraduate Science and Engineering. National Academies Press (2012)
62. Solomon, C., Harvey, B., Kahn, K., Lieberman, H., Miller, M.L., Minsky, M., Papert, A., Silverman, B.: History of Logo. Proceedings of the ACM on Programming Languages **4**(HOPL), 1–66 (2020)
63. Soloway, E., Ehrlich, K.: Empirical studies of programming knowledge. IEEE Transactions on software engineering **SE-10**(5), 595–609 (1984)
64. Szabo, C., Sheard, J.: Learning theories use and relationships in computing education research. ACM Transactions on Computing Education (TOCE) p. in press (2022)
65. Szabo, C., Falkner, N., Petersen, A., Bort, H., Cunningham, K., Donaldson, P., Hellas, A., Robinson, J., Sheard, J.: Review and use of learning theories within computer science education research: primer for researchers and practitioners. In: ITiCSE 2019 Working Group Reports, ITiCSE WGR 2019, pp. 89–109. Association for Computing Machinery (2019)
66. Tedre, M., Toivonen, T., Kahila, J., Vartiainen, H., Valtonen, T., Jormanainen, I., Pears, A.: Teaching Machine Learning in K–12 Classroom: Pedagogical and Technological Trajectories for Artificial Intelligence Education. IEEE Access **9**, 110558–110572 (2021). DOI https://doi.org/10.1109/ACCESS.2021.3097962. Conference Name: IEEE Access
67. Tenenberg, J., Malmi, L.: Conceptualizing and using theory in computing education research. ACM Transactions on Computing Education (2022)
68. Utting, I., Tew, A.E., McCracken, M., Thomas, L., Bouvier, D., Frye, R., Paterson, J., Caspersen, M., Kolikant, Y.B.D., Sorva, J., et al.: A fresh look at novice programmers' performance and their teachers' expectations. In: Proceedings of the ITiCSE working group reports conference on Innovation and technology in computer science education-working group reports, pp. 15–32 (2013)
69. Valentine, D.W.: CS educational research: a meta-analysis of SIGCSE technical symposium proceedings. In: SIGCSE '04: Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education, pp. 255–259. ACM Press, Norfolk, Virginia, USA (2004). DOI http://doi.acm.org/10.1145/971300.971391

70. Wahyuningsih, S., Nurjanah, N.E., Rasmani, U.E.E., Hafidah, R., Pudyaningtyas, A.R., Syamsuddin, M.M.: STEAM Learning in Early Childhood Education: A Literature Review. International Journal of Pedagogy and Teacher Education **4**(1), 33–44 (2020). DOI https://doi.org/10.20961/ijpte.v4i1.39855.
71. Weinberg, G.M.: The psychology of computer programming, vol. 29. Van Nostrand Reinhold New York (1971)
72. Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., Wilensky, U.: Defining Computational Thinking for Mathematics and Science Classrooms. Journal of Science Education and Technology (2016). DOI https://doi.org/10.1007/s10956-015-9581-5
73. Wiggberg, M., Gulliksen, J., Cajander, å., Pears, A.: Defining Digital Excellence: Requisite Skills and Policy Implications for Digital Transformation. IEEE Access **10**, 52481–52507 (2022). DOI 10.1109/ACCESS.2022.3171924. Conference Name: IEEE Access
74. Wing, J.M.: Computational thinking. Communications of the ACM **49**(3), 33–35 (2006)
75. Xie, B., Loksa, D., Nelson, G.L., Davidson, M.J., Dong, D., Kwik, H., Tan, A.H., Hwa, L., Li, M., Ko, A.J.: A theory of instruction for introductory programming skills. Computer Science Education **29**(2–3), 205–253 (2019)
76. Yang, K., Liu, X., Chen, G.: The influence of robots on students' computational thinking: A literature review. International Journal of Information and Education Technology **10**(8), 5 (2020)