Essential Aspects of Gender-inclusive Computer Science Education

Lisa Sällvin, Lena-Maria Öberg and Peter Mozelius
Department of Communication, Quality Management and Information Systems, Mid Sweden University, Östersund, Sweden

lisa.sallvin@miun.se
lena-maria.oberg@miun.se
peter.mozelius@miun.se

Abstract: Computer science is a higher education domain that still show a significant male dominance. Many research studies have highlighted the importance of diversity and gender balance in computer science related areas such as software engineering and system development. However, there is still a well-identified problem that university programmes and courses on computer science fail to attract the female audience. The objective of this study is to investigate the concept of gender-inclusive computer science (CS) education with the aim of broadening participation in CS courses and programs. This is conducted through a literature study, initially focusing on keywords and research areas, and subsequently searching into existing research. The research question that guided the study was: "What concepts can be found in literature to make computer science education more gender-inclusive?". Data were analysed thematically in a two-step analysis process inspired by the grounded theory methods of Open coding and Axial coding. Findings suggest that there is significant room for learning in this field, particularly from Critical CS education studies. The Open coding analysis showed that the findings can be categorised into eight main themes. In the Axial coding the themes were merged, refined, renamed, and centred around the main axial theme of 'Epistemological pluralism'. Other essential themes that all are related to the axial main theme were: 'Design and creativity', 'Bias awareness and ethics', 'Collaboration and communication', 'Self-regulated learning', 'Real-world applications', and 'Role models and mentorship'. The result of this study is presented through a visual model that illustrates essential aspects of inclusive computer science education. The paper also proposes various directions for future research.

Keywords: Gender-inclusion, Gender-inclusive Learning and Teaching, Epistemological Pluralism, Computer Science Education, Diversity in Computer Science

1. Introduction

Girls and women are key players in designing and crafting solutions to improve our lives, and the ongoing issue of female under-representation in science, technology, engineering and mathematics (STEM) education hinders progress towards sustainable development (Chavatzia, 2017). Mathematically talented females are restricted to choose a career within STEM by cultural barriers, gender stereotypes, or misinformation (Wang & Degol, 2017). Lots of initiatives exist with a focus on role models, mentorship, networks, etc., assuming that women need to change and increase their confidence to fit into a male-dominated and masculine culture. This approach may in fact have very little impact on the gender balance and can be seen as reproducing and reinforcing the status quo (Björkman, 2005). Instead of 'fixing the women', Jansson and Sand (2021) and Björkman (2005) argue that the focus should be on 'fixing the knowledge' and challenging the norms.

According to an ecological framework presented in UNESCOs report by Chavatzia (2017), the factors impacting participation, achievement and progression of girls and women in STEM studies and careers can be categorized into four different levels: individual level, family and peer level, school level, and societal level. Factors that provide inclusive and gender-responsive STEM education can span across several levels but in this study the focus is on how to develop and deliver gender-inclusive curricula in CS education. The aim is to gain a better understanding of the barriers on education-system level, that are preventing female students from entering and progressing in computer science education. This is done by collecting existing knowledge on how to design inclusive and gender-sensitive CS education. According to Holman et al. (2018) and Bjørn et al. (2023) the gender gap in STEM will not close without further reforms and structural changes in education. Built on this backdrop, the research question to answer in the literature study was: "What concepts can be found in literature to make computer science education more gender-inclusive? ".

A description of the used methods and materials and the analysis process can be found in the following section 2. Findings from the literature search have been presented, analysed and discussed in section 3, category by category as illustrated in Figure 1. Finally, conclusions and ideas for future research were presented in section 4, followed up by a short reflection on the limitations of this scoping literature review.

2. Methodology

This study was designed as a scoping review, with the suggested idea of providing an overview of a chosen topic as described by Munn et al., 2018). The scoping review is a relevant approach when the studied topic is under researched, complex or heterogeneous (Mays, Roberts & Popay, 2001; Pham et al., 2014;). Criteria that all were present in this investigation of Inclusive CS Education area. Moreover, it is a frequent choice in literature studies aiming at definition of concepts and to identify knowledge gaps. This is a type of literature review that offers a method of finding key concepts in a selected, specific research area, and to identify the main sources of available evidence (Munn et al., 2018). A fundamental strive in this scoping review was to synthesise research results to a foundation for future research studies. Considering this aim, the research question was formulated concrete and specifically to facilitate to meet the aim. Despite the tailored and narrow research design, authors claim that the results from the conducted literature review also can add general knowledge, useful for design of Inclusive CS Education.

2.1 Database Search

After identifying a few on-topic papers that serve as a core (Turkle & Papert, 1990; Björkman, 2005 and Ko et al. 2023), relevant keywords were collected and utilized for future searches. The objective was to gather a broad spectrum of answers to form a foundation for future research. The search strategy involved employing a focused search string while conducting searches across a number of search engines with potential to find answers to the research question. The search began with Google Scholar, then extended to Web of Science and the more specialized resource the ACM Digital Library. Numerous searches led to papers presented in the ACM Technical Symposium on Computer Science Education (SIGCSE), so an additional search of their material was conducted. The following search query was used:

("COMPUTER SCIENCE") AND ("EDUCATION") AND ("GENDER") OR ("INCLUSIVE LEARNING") OR ("BROADENING PARTICIPATION") OR ("CS FOR ALL") OR (“WOMEN IN STEM”) OR (“NORM-CRITICAL PEDAGOGY”)

To be classified as a relevant finding, the paper should include examples of how computer science can be more inclusive and gender-responsive, either in higher education or in general contexts. The papers should provide one or more answers to the research question. However, vague responses were disregarded, with the focus instead on concrete examples of didactical approaches and concepts. Inclusion and exclusion criteria are listed in Table 1 below.

Table 1: Inclusion and exclusion criteria

Included	Excluded
Defining concrete education approaches related to computer science (including software engineering and programming).	Generic or vague educational concepts, not specifically related to computer science or STEM.
Research relevant to higher education or secondary school.	Studies conducted in preschool or elementary school.
Various types of studies: empirical and reviews.	
Various types of literature: conference papers, journals and books.	
Written in English or Swedish	
Available via digital libraries	

2.2 Data Analysis

This scoping review was carried out in combination with a two-step inductive thematic analysis as outlined by Braun and Clarke (2012). These two steps were guided by the Grounded theory (GT) analysis concepts of Step 1) Open coding and Step 2) Axial coding. The coding process in an inductive thematic analysis clearly resembles the process in GT analysis. As conducted in the study by Samplaski (2018), an iterative thematic analysis process could preferably be divided into the steps of Open coding and Axial coding.

2.2.1 The Open Coding

In the initial analysis phase, here referred to as Open coding, text extracts from the selected publications were broken down into units of meaning, with the idea of labelling and conceptualising data as outlined by Khandkar (2009). The open coding was initiated by the first author with the idea of finding initial codes based on found text extracts or subcodes. Later the categories were fine-tuned and renamed in discussions involving all the authors according to the concept of investigator triangulation. Investigator triangulation or triangulation analysis has been defined by Patton (2002, p. 560), as "having two or more persons independently analyse the same qualitative data and compare their findings". The coding process followed the six-phase guidelines recommended in outlined by Braun and Clarke (2012) for thematic analysis. The process for an inductive thematic analysis is very close to how the Open coding process is defined in Grounded theory. Finally, according to the sixth phase in this thematic analysis guidelines, the outcomes from the Open coding step were written up and presented separately under 'Findings' here below.

2.2.2 The Axial coding

The next phase of Axial coding was guided by the idea of "coding that treats a category as an axis around which the analyst delineates relationships and specifies the dimensions of the category" (Bryant & Charmaz, 2007, p. 603). In the same way as in the Open coding phase, the Axial coding was also carried out according to the idea of investigator triangulation approach where all the authors collaborated in a 'triangulating analysis' to revise categories from the previous step. Finally, the preliminary categories from the Open coding step were discussed collaboratively and reassembled into more abstract conceptual categories around the central axis of Epistemological pluralism.

3. Findings

The findings from the literature search are summarized in a table, followed by the results from the thematic analysis.

3.1 Search Results

Below is a list of relevant findings related to enhancing the inclusivity of computer science education, derived from literature on critical CS education.

Table 2: Literature summary of inclusive CS education

Reference	Relevant findings
Turkle, S. and Papert, S., 1990. Epistemological pluralism: Styles and voices within the computer culture. Signs: Journal of women in culture and society, 16(1), pp.128-157.	• Epistemological pluralism, multiple ways of thinking. Different styles and voices within the computer culture. Logical, linear, formal and abstract thinking (masculine) vs contextual and narrative. The planners approach vs the bricolage approach.
Björkman, C., 2005. Crossing boundaries, focusing foundations, trying translations: Feminist technoscience strategies in computer science (Doctoral dissertation, Blekinge Institute of Technology).	• How to broaden the meaning of 'knowing computer science'. This includes diversity among students, diversity in ways of knowing and learning (epistemological pluralism), diversity among practitioners of CS, and diversity of practices and approaches to knowledge in the discipline.
Amy J. Ko, Anne Beitlers, Brett Wortzman, Matt Davidson, Alannah Oleson, Mara Kirdani-Ryan, Stefania Druga, Jayne Everson, 2023. Critically Conscious Computing: Methods for Secondary Education. https://criticallyconsciouscomputing.org/, retrieved 10/3/2023.	• Weave together pedagogy about technical ideas in CS with pedagogy about ethics, morality, and social justice • Design-related activities help engage diverse groups of learners. • Framing CS as an expressive discipline that centres students' creativity and interests. like an artist having a vision for an experience, and then creating it. • Critically analyse algorithms that others have written and modify them to be more just. • Distinguish *what* to make with computing and *how* to make it but include both. E.g. by using a design process including steps from ideation to test and verification. • Many students find abstract puzzle solving cold, formal, and irrelevant. • Broadening the CS knowledge. Apart from technical CS knowledge there is social, cultural, and political CS knowledge.
Loksa, D., Ko, A.J., Jernigan, W., Oleson, A., Mendez, C.J. and Burnett, M.M., 2016. Programming, problem solving, and self-awareness: Effects of explicit guidance. In Proceedings of the 2016 CHI conference on human factors in computing systems (pp. 1449-1461).	• Promote and develop metacognitive awareness in programming. By providing a physical representation of problem-solving stages to help learners monitor their state, and prompt learners to describe their problem-solving stage when they ask for help.
Li, P.L., Ko, A.J. and Zhu, J., 2015. What makes a great software engineer?. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (Vol. 1, pp. 700-710). IEEE.	• Grading non-functional attributes of the code, such as elegance, anticipates needs, and creative. • Collaborations with non-software-engineers
Kumar, A.N. and Raj, R.K., 2022. Computer Science Curricula 2023 (CS2023) Community Engagement by the ACM/IEEE-CS/AAAI Joint Task Force. In Proceedings of the 54th ACM Technical Symposium on	• "Pair programming with AI" Large Language Models (LLMs) such as OpenAI's ChatGPT, and IDEs powered by them such as GitHub Copilot. • Introducing students to computer scientists who share their

Reference	Relevant findings
Computer Science Education V. 2 (pp. 1212-1213). https://csed.acm.org/	identities. Have peer mentors that share the identities of all your students. • Real-world projects and examples. • Connect the classroom to the outside world.
Davis, K., White, S.V., Dinah, B.C. and Scott, A., 2021. Culturally Responsive-Sustaining Computer Science Education: A Framework. Technical Report. Kapor Centre. https://www.kaporcenter.org/equitablecs	• Educators actively seek out and recruit diverse guest speakers and experts representing underrepresented or marginalized groups in computing
Medel, P. and Pournaghshband, V., 2017. Eliminating gender bias in computer science education materials. In Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education (pp. 411-416).	• Awareness of language bias. Using non stereotypical pronouns and names in the context of CS lectures and course materials.
Frieze, C. and Quesenberry, J.L., 2013. From difference to diversity: including women in the changing face of computing. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 445-450).	• Create a friendly, comfortable and hard-working atmosphere
Albusays, K., Bjorn, P., Dabbish, L., Ford, D., Murphy-Hill, E., Serebrenik, A. and Storey, M.A., 2021. The diversity crisis in software development. IEEE Software, 38(2), pp.19-25.	• Diversifying computer science stereotypes Include ethical aspects of technology design
Wang, M.T. and Degol, J.L., 2017. Gender gap in science, technology, engineering, and mathematics (STEM): Current knowledge, implications for practice, policy, and future directions. Educational psychology review, 29, pp.119-140.	• Emphasize effort and hard work instead of talent • Add more storytelling to STEM learning • Break down stereotypes about women and STEM • Provide more female role models • Eliminating non-family-friendly policies
Mozelius, P., 2018. It is getting better, a little better– female application to higher education programmes on Informatics and System science. In *proceedings of the International Conference on Gender Research* (pp. 249-254).	• Combine computer science with design • Real world applications. • Mentorship • Pair-programming • Educational programs that offer the opportunity to study remotely.
Bjørn, P., Menendez-Blanco, M. and Borsotti, V., 2023. Diversity in Computer Science: Design Artefacts for Equity and Inclusion (p. 122). Springer Nature.	• including critical approaches to computing and accessibility into the core computer science curriculum • Use examples of the impact of bias built into software.
Bergstrom, I. and Lotto, R.B., 2015. Code Bending: A new creative coding practice. Leonardo, 48(1), pp.25-31.	• Creative coding: Artistic creation through the medium of program instruction.
Chavatzia, T., 2017. Cracking the code: Girls' and women's education in science, technology, engineering and mathematics (STEM) (Vol. 253479). Paris, France: Unesco	• Teaching using cognitive activation strategies motivates and engages girls. • Align teaching to real-life experiences and interest of girls, e.g. social aspect. • Anonymous grading in male-dominated subjects is advantageous for girls. • Recruiting male and female teachers. • Removing gender bias from learning materials • Linking female students to mentorship opportunities and female role models.
Master, A., Cheryan, S. and Meltzoff, A.N., 2016. Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. Journal of educational psychology, 108(3), p.424.	• Girls were more likely to take a CS introductory course in a physical classroom environment where they felt they belonged.
Wagner, I., 1994. Connecting communities of practice: Feminism, science, and technology. In Women's Studies International Forum (Vol. 17, No. 2-3, pp. 257-265). Pergamon	• Accept the validity of multiple ways of knowing and doing (epistemological pluralism)
Sultan, U., Axell, C. and Hallström, J., 2023. Bringing girls and women into STEM?: Girls' technological activities and conceptions when participating in an all-girl technology camp. International journal of technology and design education, pp.1-25.	• Making friends with like-minded peers. • Incorporating creative activities can positively impact participants' self-efficacy.

3.2 Analysis Results

The outcome from the open coding analysis of the search findings resulted in eight preliminary categories for a more gender-inclusive CS education: Bias awareness and ethics, Collaboration and communication, Metacognitive awareness, Epistemological pluralism, Real-world applications, Creativity, Design, and finally Emotional support and mentorship. These preliminary categories were then discussed collaboratively and reassembled into more abstract conceptual categories around one central axis. This diversity of practices and approaches to knowledge is illustrated in Figure 1.

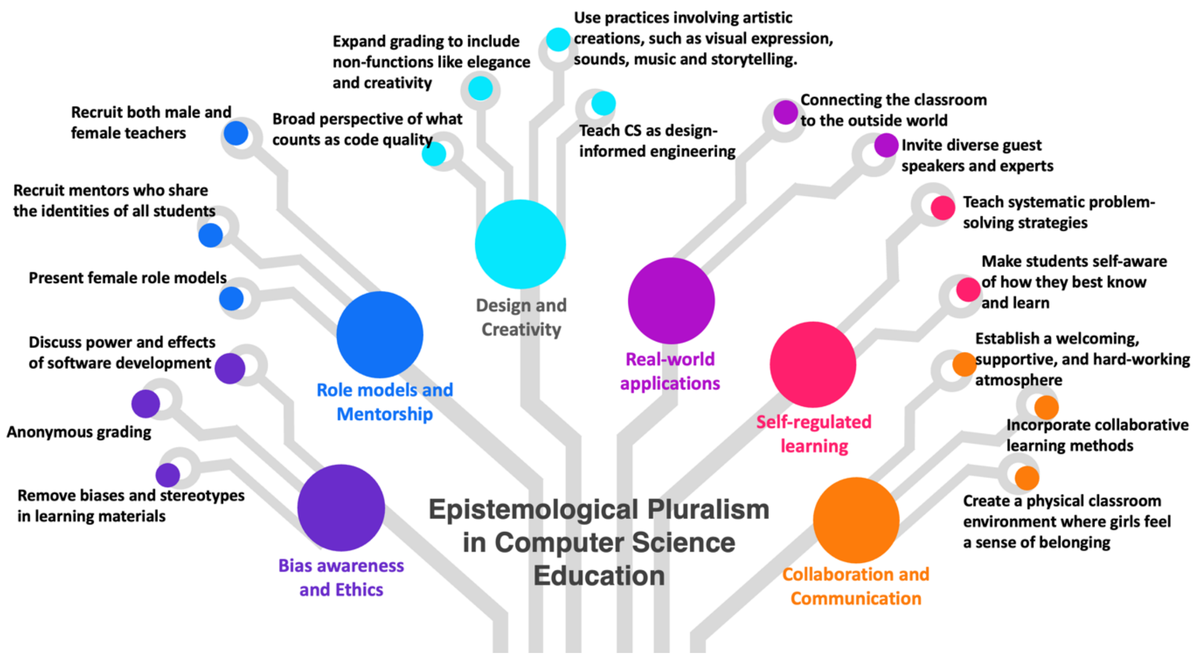

Figure 1: Visual representation of findings, with six categories centred around Epistemological pluralism in CS education.

Epistemological pluralism in CS education was found to be the central category that the other categories rely on. Björkman (2005) suggests in her Ph.D. thesis on feminist technoscience strategies in computer science that a focus on epistemological pluralism has the potential to enrich computer science education. She explored ways to broaden the concept of 'knowing computer science'. Beyond diversity among students and practitioners of CS, achieving this involves diversity in ways of knowing and learning. Epistemological pluralism is about accepting the validity of multiple ways of knowing, doing and thinking (Wagner, 1995). In the field of computer science, it can mean a diversity of practices and approaches to knowledge (Björkman, 2005).

Turkle and Papert (1990) describes the different styles and voices within the computer culture, such as logical, linear, formal, and abstract vs. contextual and narrative. There's the planners' approach vs. the bricolage approach. The bricolage approach contrasts with the classic engineering approach that typically begins with systematically developing a plan. It aligns more with tinkering, play, and the methods of an artist. The programming tool Scratch (Massachusetts Institute of Technology, 2023), developed for kids, employs this playful approach to encourage a process of testing boundaries, experimenting with new possibilities, and engaging with the world (Resnick & Rosenbaum, 2013), comparing programming to building with Lego.

The research findings in each sub-category are summarized below.

3.2.1 Bias Awareness and Ethics

There are three facets to bias awareness in CS education. One aspect involves biases and stereotypes present in the learning materials (Medel & Pournaghshband, 2017; Chavatzia, 2017). A second relates to being aware of bias, ethics, and morality when developing software and the importance of discussing the power and effects of software development (Bjørn et al.,2023; Chavatzia, 2017; Albusays et al., 2021; Ko et al., 2023). A third aspect is the bias of the teacher. Studies show that anonymous grading in male-dominated subjects is advantageous for girls (Chavatzia, 2017). All three aspects have the potential to foster inclusivity for a broader group of students. Ko et al. (2023) suggest broadening the CS knowledge to include not only technical CS knowledge but also social, cultural, and political CS knowledge.

3.2.2 Role Models and Mentorship

Taking classes or joining after-school camps is also about making friends with like-minded peers (Sultan et al., 2023). For female students who are in the minority in CS classes, it is important to have female role models, and providing opportunities for mentorship have been a successful strategy (Chavatzia, 2017; Mozelius, 2018, Wang & Degol, 2017). Kumar and Raj (2022) point out the importance of recruiting mentors who share the identities of all your students, as well as recruiting both male and female teachers (Chavatzia, 2017).

3.2.3 Design and Creativity

Combining computer science with design and using design-related activities help engage diverse groups of learners (Ko et al., 2023; Mozelius, 2018). According to Ko et al. (2023) "design and engineering is a never-ending duality and should be taught as such". CS might be taught as design or as engineering, but preferably as 'design-informed engineering'. This can be done by distinguishing *what* to make with computing and *how* to make it, but include both. For example, by using a design process during the course, including different steps from ideation to test and verification.

To incorporate creativity into the learning of programming can have a positive impact on participants' self-efficacy and may be another approach to broadening participation in CS classes (Sultan et al., 2023; Li et al., 2015). There are numerous examples of creative coding practices involving artistic creations, such as visual expression or sounds and music (Bergstrom & Lotto, 2015). To foster creativity there are ways to broaden the perspective of what counts as code quality (Ko et al., 2023), and Li et al. (2015) suggest expanding grading to include non-functional attributes of the code, such as elegance, anticipates needs, and creativity. Many students find abstract puzzle solving irrelevant (Ko et al., 2023) but Wang and Degol (2017) suggests adding more storytelling to the learning experience.

3.2.4 Real-world Applications

Integrating real-world applications into computer science education enhances the relevance and engagement of all students, with particular significance for female students (Mozelius, 2018; Kumar and Raj, 2022; Ko et al., 2023). This also exposes students to the complexities and challenges they may encounter in their future careers and prepares them for the dynamic nature of the field (Kumar and Raj, 2022). Another strategy to engage a broader spectrum of students is to actively seek for and invite diverse guest speakers and experts representing underrepresented or marginalized groups in computing (Davis et al., 2021). Kumar and Raj (2022) also suggest connecting the classroom to the outside world as a way to increase interest in CS classes.

3.2.5 Self-regulated Learning

Beyond teaching programming languages and tools, Loksa et al. (2016) argue that teaching cognitive aspects of programming is important. The learner should be aware of how she knows and learns. This involves teaching both systematic aspects, such as problem-solving strategies, and self-aware aspects, like metacognitive awareness. This can be accomplished by providing a physical representation of problem-solving stages to assist learners in monitoring their progress, and prompt learners to describe their problem-solving stage when they ask for help. Another example of how to facilitate this is by building a classroom culture around self-regulation (Ko et al., 2023). Teaching using this kind of cognitive activation strategies can particularly motivate and engage girls (Chavatzia, 2017; Loksa et al., 2016)

3.2.6 Collaboration and Communication

Establishing a welcoming, supportive, and hard-working atmosphere is one way to get more women in computer science education (Frieze & Quesenberry, 2013). A study by Master et al. (2016) indicates that creating a physical classroom environment where girls feel a sense of belonging increases their likelihood of enrolling in a CS introductory course.

Incorporating collaborative learning methods, such as pair programming, has been shown to be effective in enhancing the learning experience. Pair programming with artificial intelligence (AI) (Kumar & Raj, 2022) and traditional pair programming approaches (Mozelius, 2018) both contribute to fostering a cooperative learning environment appreciated by female students. Additionally, collaborative initiatives with non-software engineers, as suggested by Li et al. (2015), can further enrich the educational landscape and promote diversity within the field.

4. Conclusions and Future Work

Hockings (2010) define inclusive learning and teaching in higher education as the ways in which pedagogy, curricula and assessment are designed and delivered to engage students in learning that is meaningful, relevant and accessible to all. The aim of this study was to gain a better understanding of the barriers on education-system level, that are preventing female students from entering and progressing in computer science education. This has been done by searching for relevant literature that contribute to answer the research question. The findings were analysed and grouped in six categories: Design and creativity, Bias awareness and ethics, Collaboration and communication, Self-regulated learning, Real-world applications, and Role models and mentorship.

All these categories would definitely contribute to a more gender inclusive educational design centred around the concept of Epistemological pluralism. However, our choice for future research is to investigate further the categories Design and creativity, Collaboration and communication, and Self-regulated learning. Areas where the authors have earlier experiences as well as subject matter knowledge. Finally, those three categories will also be implemented and tested in design for courses in CS education.

4.1 Limitations of the Study

This study was carried out as a scoping review with the aim of exploring the barriers in the current education structure that make female students reluctant to enrol and complete university programmes on computer science. The research question was answered, and findings can be used for further research in this field. However, a scoping review do not, by definition, contain the same level of details as systematic literature reviews, and this study was based on a limited number of databases. For that reason, this study is valuable for further research on the chosen topic, but the concrete guidance for policy or more general practice is limited.

References

Albusays, K., Bjorn, P., Dabbish, L., Ford, D., Murphy-Hill, E., Serebrenik, A. and Storey, M.A., 2021. The diversity crisis in software development. IEEE Software, 38(2), pp.19-25.

Bergstrom, I. and Lotto, R.B., 2015. Code Bending: A new creative coding practice. Leonardo, 48(1), pp.25-31.

Bjørn, P., Menendez-Blanco, M. and Borsotti, V., 2023. Diversity in Computer Science: Design Artefacts for Equity and Inclusion (p. 122). Springer Nature.

Björkman, C., 2005. Crossing boundaries, focusing foundations, trying translations: Feminist technoscience strategies in computer science (Doctoral dissertation, Blekinge Institute of Technology).

Braun, V., & Clarke, V. 2012. Thematic analysis. In H. Cooper, P. M. Camic, D. L. Long, A. T. Panter, D. Rindskopf, & K. J. Sher (Eds.), *APA handbook of research methods in psychology, Vol. 2. Research designs: Quantitative, qualitative, neuropsychological, and biological* (pp. 57–71). American Psychological Association. https://doi.org/10.1037/13620-004

Bryant, A. and Charmaz, K. eds., 2007. *The Sage handbook of grounded theory*. Sage.

Chavatzia, T., 2017. Cracking the code: Girls' and women's education in science, technology, engineering and mathematics (STEM) (Vol. 253479). Paris, France: Unesco.

Davis, K., White, S.V., Dinah, B.C. and Scott, A., 2021. Culturally Responsive-Sustaining Computer Science Education: A Framework. Technical Report. Kapor Center. https://www. kaporcenter. Org/equitablecs.

Frieze, C. and Quesenberry, J.L., 2013. From difference to diversity: including women in the changing face of computing. In Proceeding of the 44th ACM technical symposium on Computer science education (pp. 445-450).

Hockings, C., 2010. Inclusive learning and teaching in higher education: A synthesis of research. York: Higher Education Academy.

Holman, L., Stuart-Fox, D. and Hauser, C.E., 2018. The gender gap in science: How long until women are equally represented?. PLoS biology, 16(4), p.e2004956.

Jansson, U. and Sand, J., 2021. Genusperspektiv på framtidens högteknologiska arbetsliv: En nordisk forskningsöversikt, inventering och analys av utbildningsval inom STEM (Science, Technology, Engineering and Mathematics).

Khandkar, S. H., 2009. Open coding. *University of Calgary*, *23*(2009).

Amy J. Ko, Anne Beitlers, Brett Wortzman, Matt Davidson, Alannah Oleson, Mara Kirdani-Ryan, Stefania Druga, Jayne Everson, 2023. *Critically Conscious Computing: Methods for Secondary Education.* https://criticallyconsciouscomputing.org/, retrieved 10/3/2023.

Kumar, A.N. and Raj, R.K., 2022. Computer Science Curricula 2023 (CS2023) Community Engagement by the ACM/IEEE-CS/AAAI Joint Task Force. In Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (pp. 1212-1213). https://csed.acm.org/

Li, P.L., Ko, A.J. and Zhu, J., 2015. What makes a great software engineer?. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering (Vol. 1, pp. 700-710). IEEE.

Loksa, D., Ko, A.J., Jernigan, W., Oleson, A., Mendez, C.J. and Burnett, M.M., 2016. Programming, problem solving, and self-awareness: Effects of explicit guidance. In Proceedings of the 2016 CHI conference on human factors in computing systems (pp. 1449-1461).

Massachusetts Institute of Technology. 2023. Scratch https://scratch.mit.edu/

Master, A., Cheryan, S. and Meltzoff, A.N., 2016. Computing whether she belongs: Stereotypes undermine girls' interest and sense of belonging in computer science. Journal of educational psychology, 108(3), p.424.

Mays, N; Roberts, E; Popay, J; 2001. *Synthesising research evidence.* In: Fulop, N; Allen, P; Clarke, A; Black, N, (eds.) Studying the organisation and delivery of health services: research methods. Routledge, London, pp. 188-220.

Medel, P. and Pournaghshband, V., 2017. Eliminating gender bias in computer science education materials. In Proceedings of the 2017 ACM SIGCSE technical symposium on computer science education (pp. 411-416).

Mozelius, P., 2018. It is getting better, a little better–female application to higher education programmes on Informatics and System science. In *proceedings of the International Conference on Gender Research* (pp. 249-254).

Munn, Z., Peters, M. D., Stern, C., Tufanaru, C., McArthur, A., & Aromataris, E. 2018. Systematic review or scoping review? Guidance for authors when choosing between a systematic or scoping review approach. *BMC medical research methodology*, *18*(1), 1-7

Patton, M. Q. 2001. *Qualitative research & evaluation methods*. 3rd edition, Sage publications.

Pham, M. T., Rajić, A., Greig, J. D., Sargeant, J. M., Papadopoulos, A., & McEwen, S. A. 2014. A scoping review of scoping reviews: advancing the approach and enhancing the consistency. *Research synthesis methods*, *5*(4), 371-385.

Resnick, M. and Rosenbaum, E., 2013. Designing for tinkerability. *Design, make, play: Growing the next generation of STEM innovators*, pp.163-181.

Samplaski, M. K. 2018. A thematic analysis of online discussion boards for vasectomy. *Urology*, *111*, 32-38.

Sultan, U., Axell, C. and Hallström, J., 2023. Bringing girls and women into STEM?: Girls' technological activities and conceptions when participating in an all-girl technology camp. International journal of technology and design education, pp.1-25.

Turkle, S. and Papert, S., 1990. Epistemological pluralism: Styles and voices within the computer culture. Signs: Journal of women in culture and society, 16(1), pp.128-157.

Wagner, I., 1994. Connecting communities of practice: Feminism, science, and technology. In Women's Studies International Forum (Vol. 17, No. 2-3, pp. 257-265). Pergamon.

Wang, M.T. and Degol, J.L., 2017. Gender gap in science, technology, engineering, and mathematics (STEM): Current knowledge, implications for practice, policy, and future directions. Educational psychology review, 29, pp.119-140.