T I N K E R

AN AUTHENTIC LEARNING & GENDER INCLUSIVE FRAMEWORK FOR TEACHING INFORMATICS IN SCHOOLS ACROSS EUROPE

WP2 / D2.2

TINKER Framework and Toolkit

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor the granting authority can be held responsible for them. Project Number: 101132887

Please cite this publication as:

TINKER project (2024). *WP2 / D2.2 TINKER Framework and Toolkit*. Available at https://tinker-project.eu/resources/framework-and-toolkit/

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor the granting authority can be held responsible for them. Project Number: 101132887

Co-funded by the European Union

AN AUTHENTIC LEARNING & GENDER INCLUSIVE FRAMEWORK FOR TEACHING INFORMATICS IN SCHOOLS ACROSS EUROPE

Table of Contents

1. Introduction

One of the main objectives of "WP2: A Framework and Toolkit for Informatics Education", of the TINKER project is to **develop a framework to teach and assess informatics in an authentic and gender-inclusive way** in upper primary and lower secondary education (10-14 years old). This proposed framework will serve as a basis for the development of a relevant Toolkit with age-appropriate learning scenarios for teaching and assessing informatics competencies. In this way, teachers, as the primary target group, will be equipped with the necessary skills to apply this pedagogical framework and design learning scenarios for classroom use.

Specifically, the proposed framework will:

a. be based on the two previous notions of authentic learning and gender-inclusive approaches, so that countries can either use it per se or be informed by that for the (re)design of informatics curricula in upper primary and lower secondary education.
b. be based on the Informatics Reference Framework for School[1] (which outlines shared informatics competencies and learning outcomes) supporting the attempt of the developers, the Informatics for All coalition, to have a consistent European vision of informatics.
c. promote the pedagogy to develop shared competencies, taught either in a separate subject or integrated way - this way, it can be used as it is or as a reference, adapted to national contexts with different needs (e.g., exploit the principles when teaching differentiated competencies or content outlined in national curricula).
d. be the basis for developing and providing teachers with training, in the form of micro-credentials, to prepare and qualify teachers (specialists and generalists) to teach informatics.
e. pave the way for new training programmes and revised informatics curricula, potentially establishing new departments, Post-Graduate and Post-Doctoral programmes, and teacher training curricula (i.e., involvement of Ministries and universities in the partnership).

Therefore, the pedagogical framework will map the following:

[1] https://www.informaticsforall.org/the-informatics-reference-framework-for-school-release-february-2022/

a. informatics topic areas, with learning outcomes tailored to the age levels (based on and adapting the Informatics Reference Framework for School) and the recommendations of the JRC 2022 report "Reviewing Computational Thinking in Compulsory Education".

b. principles of authentic learning in the context of informatics teaching and assessment.

c. gender-inclusive practices when teaching and assessing informatics.

2. The TINKER Framework

The TINKER Pedagogical Framework was initially inspired by three key pillars: Informatics Areas & Competencies, Authentic Learning, and Gender-inclusive Practices, as described in Figure 1 below:

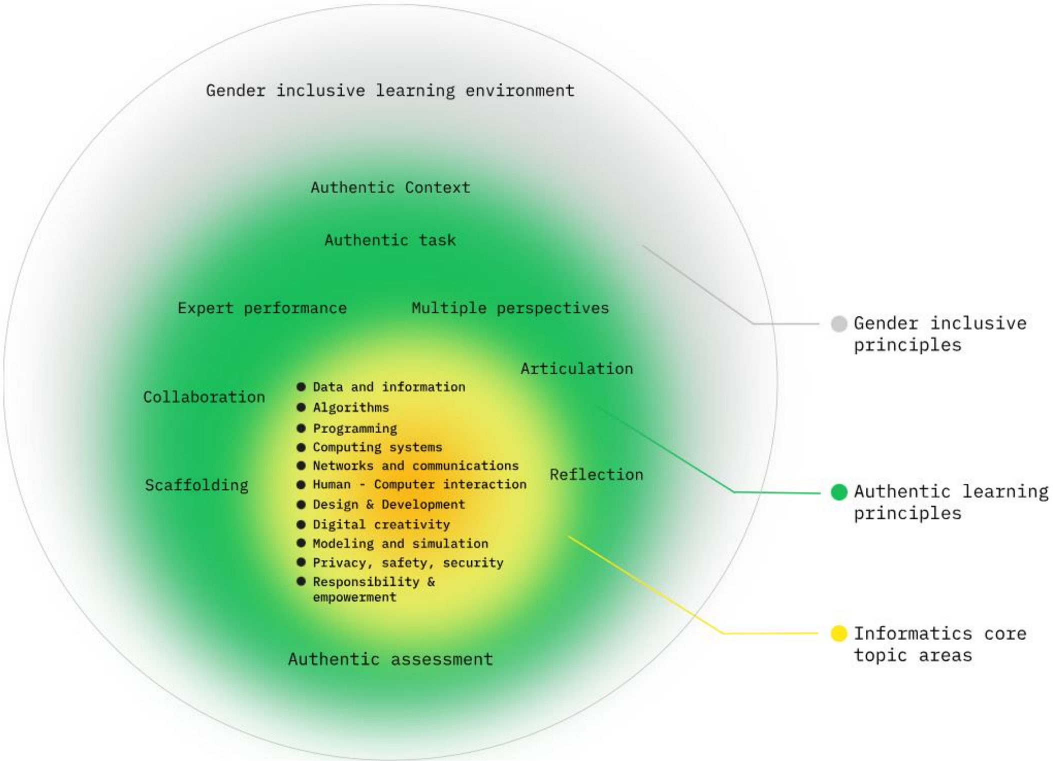

Figure 1. *The conceptual framework of TINKER*

The framework above was developed in response to the pressing challenges identified. Specifically, given the fragmentation and inconsistency in informatics curricula across European schools (Committee on European Computing Education, 2017), the lack of adequate understanding among students (European Commission, 2019), and the persistent gender imbalance in the field (Eurostat, 2021), it became evident that a more comprehensive, inclusive, and student-centred approach was needed. Additionally, the framework emphasises the importance of a balanced and authentic approach to informatics education, fostering all students' interest in an interdisciplinary way. As noted by the European Commission (2022), a balance between theory and practice, abstract concepts, and technological aspects is crucial for effective learning. By integrating these elements, the framework aims to provide a well-rounded and engaging educational experience for all students.

The findings from the compiled report across partner countries (TINKER, 2024) provided crucial evidence to inform the development of the final TINKER Pedagogical Framework. The compiled report reveals the challenges teachers face in implementing the three core pillars, emphasising the necessity for enhanced teacher professional learning. The revised framework is the result of validating the initial three main pillars and identifying an additional pillar based on the desk and field research conducted in partner countries. The proposed fourth pillar, *Teacher Professional Learning*, focuses on equipping teachers with the necessary knowledge, skills, and resources to effectively implement the TINKER Pedagogical Framework.

The TINKER Pedagogical Framework (see Figure 2) adopts a comprehensive approach to informatics education by integrating four key pillars:

A. **Informatics Areas & Competencies;**
B. **Authentic Learning;**
C. **Gender Inclusion; and**
D. **Teacher Professional Learning.**

Based on the results of the desk research and the challenges identified, these four pillars form the foundation of the framework and learning scenarios development. The four pillars represent a comprehensive approach that includes pedagogies, competencies, teacher learning and practices that focus on gender. This approach provides a clear roadmap for schools to implement effective and engaging informatics education programmes that prepare all students for a future driven by technology. These pillars, represented as overlapping circles in the diagram, symbolise their interconnected nature and mutual influence. Central to the TINKER framework is the cyclical process of **monitoring and evaluation process**, which is embedded throughout all pillars. By systematically gathering and analysing data regarding the informatics areas and competencies as well as the gender inclusion and authentic learning practices, teachers can refine teaching practices, ensuring the framework remains responsive to the evolving needs of students and the field of informatics.

Each pillar supports and reinforces the goals of the others, creating a holistic framework. For instance, an engaging real-world activity (Pillar B. Authentic Learning) can develop a specific informatics competency (Pillar A. Informatics Areas and Competencies). This activity would be designed with gender-inclusive practices in mind (Pillar C. Gender Inclusion) and consider the teacher's pedagogical knowledge and skills (Pillar D. Teacher Professional Learning), all while being informed by continuous monitoring and evaluation data. This interconnectedness fosters a comprehensive approach to informatics education, allowing for the creation of learning scenarios that are not only academically rigorous but also relevant, engaging, equitable, and adaptable.

By integrating these elements, the TINKER framework aims to empower teachers and create engaging, relevant, and inclusive learning environments that cater to the diverse needs of all students. Figure 2 below presents the described TINKER framework. Each framework's pillar will be further analysed in the sections below.

Figure 2. *TINKER Pedagogical Framework.*

The sections below explain in detail each of the framework's four main pillars.

2.1. Pillar A. Informatics Areas & Competencies

The TINKER framework addresses the need for a unified approach to informatics competencies. It leverages the Informatics4All coalition's reference framework, which outlines core areas and learning outcomes for primary and secondary schools. These areas include the following:

- Data and Information
- Algorithms
- Programming
- Computing Systems
- Networks and Communication
- Human-computer Interaction
- Design and Development
- Digital Creativity
- Modelling and Simulation
- Privacy, Safety, and Security
- Responsibility and Empowerment

This Informatics4All framework stems from an analysis of EU curricula, examining how informatics is taught across different educational levels. For example, in primary education, informatics is a separate subject in Greece and Croatia, while it is integrated into other subjects in Cyprus, and focused on digital competencies in countries like Italy and the Netherlands. In secondary education, informatics is a compulsory subject in Cyprus, and Greece, an optional subject in Ireland, and integrated into other subjects in Italy. Table 1 below showcases the diverse approaches to informatics education across Cyprus, Greece, Ireland, the Netherlands, Croatia and Italy.

Table 1. Approaches to informatics education in partners' countries.

Country	Informatics as a subject	Topic Areas	Learning Outcomes (Age Specific)
Cyprus	Taught separately in lower secondary education. Integrated into design and technology, life education, mathematics, and physical sciences courses in primary education.	Basic concepts of informatics, computer hardware, operating systems, application software, networks and the Internet, cyberbullying, databases and systems analysis, algorithms, programming, and modern computer applications. Focus on specific applications like computer publishing, website development, and database management.	*Upper primary: Effective information search, basic programming, data use, responsible digital citizenship, cultivation of students' digital competence, technological literacy. * Lower Secondary: Develop problem-solving skills using computers, understand algorithms and computer programmes, cultivate critical and creative thinking.
Greece	Taught as part of the National Curriculum in upper primary and lower secondary education.	Familiarity with computers, computer software, internet services, creation and expression tools, and understanding the role of computers in daily life.	*Upper primary: Effective information search, basic programming, data use, responsible digital citizenship. Use ICT tools for creating and editing various types of content, navigate the internet safely and effectively, analyse and solve basic ICT-related problems, and apply ICT skills in real-life situations and interdisciplinary projects. * Lower Secondary: Develop problem-solving skills using computers, understand algorithms and computer programmes, cultivate critical and creative thinking. Further develop IT skills.

Country	Informatics as a subject	Topic Areas	Learning Outcomes (Age Specific)
Netherlands	Not a separate subject, but digital literacy is incorporated into the curriculum of other subjects (especially in secondary education).	Digital literacy (conscious, responsible, critical, and creative use of ICT, Digital systems, Digital media and information, Safety & privacy, Data, Artificial Intelligence (AI), Creating with digital technology, Programming, Digital technology, yourself, and the other, Digital technology, society, and the world)	Digital Literacy Curriculum Goals (Upper Primary & Lower Secondary): * Practical knowledge & skills: Using digital systems functionally, navigating the digital media and information landscape, handling digital systems, data, and privacy safely, exploring data and data processing, and exploring how AI systems work. * Design and create: Creating with digital technology and programming a computer programme using computational thinking strategies. * Interaction between digital technology, digital media, people and society: Making well-considered choices when using digital technology and digital media, and exploring how digital technology, digital media, and society interact.
Croatia	Compulsory subject in grades 5 & 6 (student's age 12-13) and optional subject in grades 7 & 8 (student's age 14-15)	Basic understanding of information and digital technology, programming, using digital tools for communication and collaboration, and understanding of personal data protection.	Ages 12-15: Apply computer technology in problem-solving, create and manage digital content and footprints, and participate in a digital society.

Country	Informatics as a subject	Topic Areas	Learning Outcomes (Age Specific)
Italy	Not a separate subject, but it is foreseen as transversal to all subjects and is highly focused on digital competence.	Digital competence, computational thinking, digital tools and technologies, programming concepts, fundamental software, spreadsheets, word processors.	Master the use of ICT and develop critical skills for their use, be able to produce simple models or graphical representations of their work using elements of technical drawing or multimedia tools, recognise and document the principal functions of a new informatics application, use internet to find the necessary information, develop programming and problem-solving skills.
Ireland	Not a standalone subject in primary or secondary education, but informatics concepts are integrated into various subjects.	* Primary Cycle (Stage 4 - Grades 5 & 6) - Upper primary level: Algorithms, hardware/software basics, internet safety, data representation. * Junior Cycle - Lower secondary level: Programming concepts, data structures & algorithms, cybersecurity, digital ethics, computer networks, coding, digital media literacy.	* Upper primary level: Basic computational thinking, simple programming, digital tool use, data handling, developing their problem-solving skills. * Lower secondary level: Programming proficiency, data structures & algorithms understanding, cybersecurity awareness, digital ethics, collaboration using digital tools.

This analysis of informatics education across the partners' countries, reveals a strong foundation in core areas like data, algorithms, and programming. Countries like Ireland demonstrate successful integration of these concepts into various subjects during primary education. However, based on the findings of the Transnational Report there is potential to further expand the scope of informatics education based on the Informatics4All Framework. Specifically, in primary education, more basic concepts of networks & communication and human-computer interaction could be introduced in a way appropriate for young students. At the same time, in lower secondary education modelling & simulation concepts could be further explored to enhance problem-solving and analytical skills.

As students progress from upper primary to lower secondary, the focus shifts from foundational skills like word processing and digital literacy to more advanced concepts such as programming, computational thinking, and digital citizenship. This reflects a gradual increase in complexity, as well as a move towards more independent learning and critical thinking. In primary education, students engage with basic digital tools and concepts in a concrete and accessible manner. As they transition into secondary education, the curriculum introduces more abstract and sophisticated topics like algorithms, data management, and problem-solving through coding and system design (European Commission, 2022; Informatics4All, 2022). The progression also includes an increasing emphasis on ethical concerns, such as privacy, data security, and the societal impact of digital technologies (Informatics4All, 2022).

Differentiated instruction plays a crucial role in supporting this progression by tailoring teaching methods to meet diverse student needs. This approach provides students with opportunities to explore informatics concepts at their own pace, enabling both mastery of foundational skills and exploration of more complex issues. In secondary education, students are encouraged to apply their knowledge in a variety of contexts, fostering essential skills such as critical thinking, collaboration, and creative problem-solving. By focusing on a solid foundation in primary education and gradually introducing more advanced topics in secondary education, teachers ensure that students not only develop a comprehensive understanding of informatics but are also well-prepared for the demands of the digital world (European Commission, 2022; Informatics4All, 2022).

TINKER enhances this framework with principles of authentic learning and gender-inclusive practices, aiming to teach and assess competencies both separately and integratively.

2.2. Pillar B. Authentic Learning

The TINKER framework adopts an authentic learning model, which emphasises real-world problem-solving and the application of knowledge in practical contexts. According to Cole (1990) and Herrington and Oliver (2000), students often view knowledge as merely educational unless it is contextually applied. The principles required for designing authentic learning environments (Herrington et al., 2014; Herrington & Oliver, 2000) are as follows:

- **Authentic context**: a virtual or physical environment reflecting how knowledge is used in real life, without simplifying things, motivating learning.

- **Example 1**: To understand databases, the students work and solve common school-related issues such as the organisation and management of data for their school library (e.g., catalogue and borrowing system), the homework and assessment assignment process (e.g., reminding tasks and deadlines) or the attendance procedure (e.g., maintaining attendance records in a database). Such contexts reflect the real-world, asking students to use their knowledge and skills to improve their daily lives as students and young citizens.

- **Authentic task**: complex tasks (not predefined steps students have to follow), with real-world relevance, are interdisciplinary, require production (not reproduction) but cannot be solved on the spot (sustained investigation over a period).
  - **Example 1**: The students are tasked to create a digital library management system for their school that keeps records of books, borrowing, and returns processes. Students must design the database to store book details, borrower information, and borrowing history. The task is open-ended as it is up to the students to decide what steps to follow to solve it; they have to decide on what data to include, how to structure and implement the system using relevant tools for spreadsheet and scripting (e.g., Google Sheets with Google Apps Script). This requires students to engage with database concepts such as storing and organising data, basic programming, automating some library tasks, and addressing usability for users such as students, teachers, and librarians. Sustained investigation is part of the task since students will have to work over several weeks to design the database, gather user requirements, test and improve based on feedback.
  - **Example 2**: The students are tasked with designing and programming a robot using a platform like LEGO WeDo 2.0 or a similar educational robotics kit to solve a real-world environmental problem, such as sorting recyclable materials. They must build and program the robot (e.g., a truck) to sort recyclable objects.

- **Expert performance (role models)**: access to expertise, seeing how experts think and work, observing real-life episodes, and having opportunities to share stories.
  - **Example 1**: A robotics expert is invited to showcase how robots can be programmed to solve environmental challenges such as recycling. The expert can show a robot

sorting different materials and explain how data is used to appropriately classify and direct each item.

- **Multiple perspectives**: the opportunity to adopt various roles and see things from different points of view.
  - **Example 1**: Students are tasked with designing a digital library management system that helps manage books, users' records, and borrowing and returning processes. The school has multiple users of such a system, each having different needs. For instance: (a) the students are the direct users who need an easy-to-use interface to search for books, confirm availability, borrow books and manage their borrowing history; (b) the librarians, as managers, need to catalogue new books, manage and update the database, track borrowing process and returns (e.g., send notifications); and (c) the teachers need to see which books are available, create reading lists and recommend books.

    The students need to identify these needs by holding discussions with their classmates, librarians and teachers, before creating the digital system.
- **Collaboration**: tasks addressed to groups, for individuals to work in pairs or teams, striving for whole-team success.
  - **Example 1**: To design the digital library system for their school, the students are divided into teams, undertaking different roles. For example, one team works on understanding the librarian tasks/work (and, in general, the library information system), one on designing the interface (three user interfaces for each different stakeholder), and one on the user experience and testing. The students can also be divided into larger groups, each designing their system, to compare each other later on.
- **Articulation**: the opportunity to articulate thoughts and results, present publicly an argument and reach understanding through social interaction.
  - **Example 1:** The students have to present their part of the school library management system, such as the database structure or user interface, to their peers and teachers. For example, the user interface team must explain the logic behind how they designed

the interface based on the different users' needs. The students present their work using diagrams and demos.

- **Reflection**: the opportunity to think about, reflect and discuss choices either in action (while making choices) or after action (after decision-making)- reflection is a social process, too.
    - **Example 1:** Throughout the development of the school library management system, students keep reflective journals, writing about any challenges they face and how they approached them, any concerns they had, what decisions they made, and how they solved unexpected issues. For example, they might write which data fields they included to balance complexity with usability. While working, the teacher poses additional reflective questions to reveal their thinking and push them to progress, finding their solutions, when challenges arise. The students write a report at the end of the project, reflecting on the whole experience (e.g., efficiency or what could be improved, collaboration with peers).
- **Scaffolding**: assistance and coaching, guidance that activates metacognition.
    - **Example:** At the beginning, the teacher gives an example of a small book database and explains how to organise information into fields (such as title, author, and genre). Students then start creating their version by elaborating on the given example to fit the needs of the school's library. They are also given structured templates with questions on key issues they need to consider (e.g., "What types of data will we need to track?", "How will users interact with the database?", "What do the users need to see?"). The teacher might also help students add the basic features to the system (e.g., add books, etc.). As students become more comfortable, gradually, the help is removed.
- **Authentic assessment**: different types of assessment are integrated, ranging from skill-based to performance-based, rather than a separate function where students use a wide range of skills, showcasing performance or products to be evaluated with proper criteria (aligned with the task).
    - **Example 1**: The students are assessed based on the process and outcome. Specifically, they are assessed based on how functional the final system is (e.g., effectiveness, user

interface quality) and how they worked collaboratively to reach this goal. They are asked to test the system with a group of end users while preparing a manual for them on how to use and manage the system. This manual is a documentation of the whole process. Thus, it shows whether they understand the system and can easily communicate technical information to non-technical users (e.g., librarians, classmates, and teachers). As part of the final demonstration, unexpected issues might arise for them to troubleshoot, assessing their understanding of the concepts. They can present their prototypes to an audience or a steering committee of experts, as described in point "expert performance".

The authentic learning model contrasts with traditional rote memorisation approaches, instead fostering deep understanding through constructivism (Piaget, 1975) and social constructivism (Vygotsky, 1978), situated learning (Lave & Wenger, 1991; Lave, 1988), communities of practice (Stein et al., 2004) and learning communities (Scardamalia & Bereiter, 1994).

When outlining the principles of authentic learning, a progression from primary to lower secondary education should be also documented. For example, teachers could consider the following:

- Authentic Tasks and Activities: Authentic learning emphasises the use of complex, ill-defined tasks that reflect real-world situations. For younger students (primary), tasks might start as discovery-based activities, allowing them to explore topics with some structure. As they progress to secondary education, tasks become more complex, requiring students to directly engage with abstract concepts such as informatics, encouraging deeper problem-solving and critical thinking skills.

- Scaffolding: Early stages of education involve significant scaffolding where teachers support the students' learning processes. As students move toward secondary education, they become more autonomous in their learning. Authentic learning environments support this transition by fostering independent thinking, particularly through engagement with increasingly abstract and discipline-specific content.

2.3. Pillar C. Gender Inclusion

The TINKER Framework adopts gender-inclusive practices informed by critical theory and pedagogy, feminist pedagogy, and intersectionality (McClure, 2000; Crenshaw, 1989). It aims to promote awareness about gender diversity, assess gender bias, balance educational activities, use gender-inclusive language, provide accessible examples, and encourage open discussions. The goal is to foster motivation for computer science among all students, focusing on girls and gender minorities, aligning with the authentic learning model.

Interest in STEM

Recently, primary and secondary schools have been working hard to integrate informatics into their curriculum, recognising that future challenges will increasingly require programming and algorithmic thinking skills. However, it has been recorded that girls tend to lose interest in STEM subjects as they get older, with lower participation already evident in secondary education (Chan, 2022).

Interest in computer science and informatics is particularly low among girls and non-binary students compared to other STEM fields (Ren, 2022), and it seems to decrease more rapidly in girls at the beginning of secondary school (around 11 - 12 years old) with minimal recovery in later education stages (De Wit et al., 2023; Happe & Buhnova, 2018; Main & Schimpf, 2017). The first barrier for girls often arises at school, where they frequently believe they are less capable and experienced than boys, leading them to dismiss informatics as unappealing. This attitude stems from deep-rooted gender stereotypes in society and family, which frame informatics as a male-dominated field (Brett, 2022; De Witt & Archer, 2015). Young women's barriers to pursuing informatics education and careers are also tied to classroom dynamics (Szláv, 2021). Factors such as teacher and peer attitudes, the curriculum, the learning environment, and a lack of knowledge about the variety of IT jobs and the skills they require contribute to the significantly low number of girls engaged and interested in informatics and computer science (Happe et al., 2021). These factors negatively affect girls' perceptions of their own STEM abilities and career aspirations, reduce their interest and confidence in STEM study, impact their ability to envision themselves in STEM fields, undermine their motivation to pursue STEM-related opportunities, and hinder the long-term engagement of girls and women in STEM (Garriot et al., 2017).

Schools' gender ideologies

Schools and classrooms are characterised by a "hidden curriculum" —unspoken norms, values, and expectations— that can influence students' behaviours, attitudes, and academic paths, often perpetuating gender biases (Gordon, 1982). Gender ideology in schools can impact students' competence beliefs, preferences, and future career motivation (Vleuten et al., 2016). Girls face significant challenges in computer science and informatics education due to limited access, negative perceptions of scientific and technological disciplines, and stereotypes that label it as a "nerdy" field (Washington et al., 2019). One of the key elements is the teacher's knowledge, beliefs, and (unconscious) biases. Teachers can contribute significantly to the gender gap in STEM education and more in computer science education, shaping the experiences and aspirations of students from an early age and with a long-term impact (Lavy & Megalokonomou, 2023; Msambwa et al., 2023). This gap is not just a matter of individual preferences; it reflects broader cultural and educational influences that begin in the classroom. When teachers exhibit different expectations for boys and girls—whether through role-modelling, biassed grading, variations in classroom engagement, or the types of feedback given—they reinforce the stereotype that computer science and related STEM fields are more suitable for males (Muntoni et al, 2019). For example, boys are more often asked challenging questions or encouraged to solve problems independently, while girls are more often given guided assistance or praised for effort rather than ability. This reinforces the notion that boys are more naturally suited for problem-solving in technical fields (Muntoni et al, 2019).

These biases can significantly affect students' self-perception and confidence. Girls who perceive a lack of encouragement or notice differences in how they are treated compared to their male peers may be less likely to pursue computer science, viewing it as a field where they do not belong or cannot excel (Msambwa et al., 2023). The cumulative effect of these experiences can dissuade girls from enrolling in advanced computer science courses or considering careers in technology. Furthermore, teachers' biases or lack of knowledge can affect how parents and peers view girls' potential in STEM, perpetuating the cycle of underrepresentation. Addressing these issues requires more than just interventions in computer science classes; a holistic approach across all subjects is needed to change the underlying gender narratives within schools. These are based on the principles of critical theory and pedagogy, which examines power relations in the classroom; feminist pedagogy, which posits that gender affects what is taught and how (McClure, 2000) as well as intersectionality, which holds the position that the intersection of multiple identities, including gender, might create discrimination (Crenshaw, 1989).

What needs to be done?

Once in computer science and informatics class, girls and non-binary students easily encounter additional barriers within a male-dominated classroom, such as gender stereotypes and sexist behaviours from peers and teachers, lack of peer support, and power imbalances (Malazita & Resetar, 2019). These hostile environments often make underrepresented students feel unsafe, isolated, or devalued, leading to decreased engagement and higher dropout rates in computing discipline and careers (Eagly, 2021). To address these issues, schools need to adopt a whole-school approach that tackles gender bias and stereotypes that push girls away from STEM subjects. This approach should also address harassment and bullying of all students to create a safe and supportive learning environment. As emphasised by the European Commission's report[2], "Gender Equality Strategy 2020-2025," a whole-school approach is crucial for fostering gender equality and challenging gender stereotypes. It involves creating a positive school climate, providing gender-sensitive teaching and learning materials, and offering training for teachers and staff to promote gender equality.

In school subjects such as STEM, gender-inclusive practices assess gender bias, create awareness about gender diversity, balance education activities, use gender-inclusive language, provide accessible examples (e.g., female role models), open discussions about gender norms and follow experiential learning pedagogy (Christou et al., 2022). By implementing reflective practices and training on gender-inclusive teaching strategies, teachers can become more aware of their unconscious biases and actively work to create a classroom environment where all students, regardless of gender, are equally encouraged and supported. Additionally, schools should create curricula that incorporate social impact projects or interdisciplinary approaches that can make computer science more appealing by showcasing its relevance to broader career paths and societal challenges. The inclusion of more foundational computer science courses and equitable access to resources (e.g., Code.org, 2024) are also essential, as data indicates that disparities in access significantly limit the participation of underrepresented groups in computer science education (Allen & Eisenhart, 2017). Moreover, schools should actively foster supportive environments by organising clubs, mentorship programmes, and events that encourage girls and non-binary students to explore technology in a collaborative and welcoming setting. For example, attending diversity-focused tech events like the “Women in Code Festival” has been shown to inspire students by connecting them with mentors and role models who share similar backgrounds (Allen & Eisenhart, 2017). Schools should also provide career guidance for all students, regardless of gender, and should emphasise the diverse applications of IT and the wide range of potential career paths available. Educating them on how IT skills can be used in various fields helps them envision a future where they actively engage in technology-related careers, increasing their interest and motivation to pursue IT at higher levels. When schools create a supportive

[2] https://data.europa.eu/doi/10.2766/915001

environment that encourages all students to explore further education and careers in tech, it reinforces their confidence and commitment, making learning more meaningful and inspiring them to stay engaged in these subjects (Kuteesa et al., 2024).

These changes can create a more inclusive culture that promotes sustained young girls and non-binary people's interest and participation in computer science and informatics. Professional development for teachers is also crucial, as teachers play a significant role in shaping students' perceptions of STEM fields. Training should focus on gender-inclusive practices to avoid biases, such as giving more challenging tasks to boys or discouraging girls from pursuing advanced studies in computer science. Research has shown that teachers' beliefs can influence students' confidence and interest in pursuing STEM careers (Demirkol et al., 2022).

Combating stereotypes in informatics

In the computing discipline, this involves using strategies that combat stereotypes and stimulate interest, particularly for underrepresented groups. Active learning and diverse didactical approaches will help narrow the gender gap in computer science education and contribute to a more diverse and inclusive field (Ren, 2022). One effective strategy is to **highlight the social impact of computer science and its interdisciplinary nature**, making the subject more relatable and appealing to students who might not initially view themselves as fitting into the traditional computer science mold. Additionally, **exposure to computing through both offline and online activities**—such as discussions, reflections, digital games, and visual programming—can foster interest in the field (Happe et al., 2021). These activities offer varied entry points into computing, making the subject more accessible to diverse students. Building self-confidence in students is another key aspect of gender-inclusive pedagogy. **Creating low-stakes opportunities for success, encouraging a growth mindset, and offering self-directed projects** are strategies that can help students—especially girls and gender minorities—feel more competent and empowered in their learning (Dweck, 2006). For instance, initial low-pressure tasks, followed by progressively more challenging activities, allow students to see their growth and abilities over time, fostering sustained interest and reducing the fear of failure. Furthermore, targeted career guidance is essential for inspiring girls to pursue IT careers (Allen & Eisenhart, 2017). By highlighting the diverse range of applications of IT, from web development and software engineering to data science and cybersecurity, teachers can broaden students' perspectives and foster a sense of future possibilities. Encouraging girls to take advanced IT courses, participate in extracurricular activities, and engage in coding competitions can further cultivate their passion and equip them with the necessary skills to excel in these fields.

Experiential learning approaches

Happe et al. (2021) argue that an effective teaching approach that nurtures students'—especially girls' and gender minorities'—motivation for computer science can follow a **cyclical model: initial contact with the subject, interest stimulation, and interest sustainability**. This aligns well with the principles of experiential learning (Christou et al., 2022), which emphasises learning through experience and reflection, allowing students to apply knowledge in real-world scenarios. Experiential learning not only reinforces concepts but also builds a deeper connection to the subject matter, particularly when activities are designed to be authentic and personally meaningful to the students.

Tinkering approach

The tinkering approach is grounded in this theoretical framework, emphasising a **self-directed and playful exploration of materials**. It often begins with an open-ended investigation, where students explore without a specific goal. Over time, they define personal objectives, conduct experiments to achieve these goals, and then make observations and interpretations based on the outcomes. This reflective process informs the selection of new goals. Through an iterative cycle of experimentation and interpretation, the tinkerer engages with the materials in a toolbox, continuously testing and refining ideas to make the system work (Resnick & Rosenbaum, 2013).

Game-based approach

As noted by Gee et al. (2020), game-based education is another promising approach well anchored in tinkering and aligned with the authentic learning model and with a high engagement potential on girls. Digital and analog games are an effective way to introduce young girls to computer science, particularly the fundamentals of programming and computational thinking (Harteveld et al., 2014). A well-designed game **allows all students to learn new concepts in an alternative classroom environment, fostering social interaction, changing the traditional teaching method, and engaging students through an activity in which all can participate**. Games offer the potential to provide experiences that promote a deeper understanding of foundational informatics concepts and their applications. Additionally, game design itself is a collaborative activity that encourages students to apply computational thinking and informatics knowledge, along with artistic and narrative elements, to create their own games.

The principles required for designing gender-inclusive environments are as follows (following the example in section 2.2, on creating a digital library management system for school, we also provide relevant examples for each of the gender inclusion principles):

- **More Diverse Role Models in STEM:** the integration of non-binary, and female role models and relatable figures who have succeeded in STEM fields into the curriculum.
  - **Example:** Invite a female data scientist or librarian to share her experience in designing and managing large library databases, showing how women are contributing to real-world STEM applications. Discussions on the historical role of women and non-binary people in STEM, and why their contributions are often overlooked, should be introduced in school to help students understand the roots of the underrepresentation of some people in science. This awareness can help address existing biases and promote a more inclusive view of scientific fields.
- **Gender-Inclusive Learning Materials & Language:** the use of gender-inclusive language and materials, regardless of gender identity (for example, the use of "they" instead of "he/she"). Assumptions about students' genders should be avoided. The use of each student's correct pronouns should be encouraged, and transphobic or biassed language in the classroom must be addressed. For additional resources, teachers can consult the Inclusion Schools Toolkit by Brighton & Hove City Council and the University of Warwick's guide on Gender-Inclusive Teaching Practices on practical tools and strategies for building a supportive, inclusive classroom culture.
  - **Example:** While developing the library management system, teachers should ensure that guidelines use gender-neutral terms, like "they" instead of "he/she," and avoid assuming the gender of users. Similarly, students should ensure that the database user interfaces also use gender-neutral terms, such as "users" or "students," instead of "he" or "she". This will help make the system inclusive for all users. By modelling inclusive language, using correct pronouns, and challenging harmful language, teachers promote a respectful environment. Paired with discussions about diversity

in STEM, these practices help students feel recognised and valued, encouraging all to actively engage.

- **Equitable Engagement in Collaborative Learning Environments:** tasks addressed to groups where every student's contribution is valued.
    - **Example:** Divide students into mixed-gender teams to design different components of the library system, ensuring that each member's contributions are equally valued and that leadership roles are rotated fairly. The teacher should ensure that group projects and teamwork should be structured in a way that promotes equality and respects diverse perspectives.
- **Inclusive Classroom Interactions:** Teachers should aim at creating a classroom environment where all students feel valued and supported. This involves providing equal attention to all students, regardless of gender, and challenging stereotypes about girls' abilities in STEM. Positive reinforcement should be focused on effort, perseverance, and specific skills, rather than gendered stereotypes. Additionally, maintaining positive nonverbal communication, such as eye contact and supportive body language, can further create a welcoming and inclusive learning space. Teachers should also address peer pressure and bullying by creating a safe space for all students to report such incidents, teaching bystander intervention strategies, and providing support for students who may feel isolated or marginalised.
    - **Example:** During a group project on designing a problem's solution, the teacher ensures that all students have equal opportunities to contribute their ideas and skills. The teacher actively listens to all students' input, asks probing questions, and provides specific feedback on their work. By creating a supportive and inclusive environment, the teacher encourages students to take risks, experiment, and learn from their mistakes.
- **Real-World Applications of STEM:** the integration of the real-world problems into tasks or activities to improve people's lives.
    - **Example:** Demonstrate how digital library systems can improve access to educational resources in under-resourced schools, particularly benefiting girls who may have

limited access to technology and learning materials. Highlight how improving these systems helps bridge educational gaps and promotes gender equality in education.

- **Encouraging research and real-world applications of IT beyond the classroom** can empower all students to explore further studies and careers in tech fields. When teachers suggest additional resources or highlight successful diverse role models in IT, they create opportunities for all students to envision themselves in these roles.
  - **Example:** Guide interested students to research the applications of AI in environmental science or healthcare, connecting technology to topics that may resonate with them. Pairing this guidance with an in-class project, like interviewing women in STEM careers, can inspire students by directly linking academic work with career possibilities. These strategies provide insights that can encourage girls, who may be less supported at home to pursue IT fields, to explore these options actively.
- **Normalising Failure and Encouraging Perseverance:** trial and error are part of the learning; failure is a natural step in the STEM learning process, fostering resilience and a growth mindset. However, girls often have lower confidence levels in STEM subjects compared to boys. This can lead to a self-fulfilling prophecy, where girls are more likely to attribute poor performance to a lack of inherent ability rather than external factors like lack of preparation or effort. By normalising failure as a crucial part of learning, teachers can help all students to develop a growth mindset and overcome self-doubt. This approach can significantly contribute to closing the gender gap in STEM fields.
  - **Example:** Encourage all students, especially girls and gender minorities, to view mistakes in the library database design process as learning opportunities, emphasising that debugging and iteration are natural parts of programming and system design.
- **Diverse Teaching Strategies:** adaptation of teaching methods to cater to diverse learning styles (for example, using a variety of teaching materials, interactive lessons, and practical applications that appeal to different students).
  - **Example:** The teacher uses varied teaching materials—such as videos, diagrams, and hands-on activities—to ensure that all students, regardless of gender or learning style, can grasp the concepts behind database design and management.

Research shows that authentic learning and gender inclusion are strongly interconnected (Singer et al., 2020). Authentic learning requires students to collaboratively solve real-life tasks of the real-world. By showing how STEM is used to solve real-world problems, including those that disproportionately affect women, the curriculum can become more relevant and engaging. Highlighting the societal impact of STEM can draw more women into these fields. Collaborative and inclusive learning environments where every student's contribution is valued can help make STEM subjects more appealing to women (Ren, 2022; Sharpe, & Rothenberg, 2018). Hence, group projects and teamwork should be structured in a way that promotes equality and respects diverse perspectives.

2.4. Pillar D. Teacher Professional Learning

This pillar recognises that for informatics education to be authentic, inclusive, and comprehensive, teachers must be well-prepared and continuously developed. It includes:

- **Training on Informatics Areas & Competencies, Authentic Learning Methodologies and Gender-Inclusive Practices**: Equipping teachers with the knowledge of core informatics concepts and effective pedagogical approaches as well as with the strategies to integrate real-world and gender-inclusive applications into informatics lessons.
- **Resources for Effective Teaching Practices**: Providing access to exemplary teaching resources that align with the core pillars.
- **Support for Ongoing Professional Development**: Ensuring teachers receive continuous training to confidently deliver informatics education.

Teacher education and continuous professional development for in-service and teacher training for pre-service in informatics, with a focus on authentic learning and gender inclusivity, can be designed to address both the professional development needs of current teachers and the foundational training of future teachers. Therefore, teachers must be well prepared and continuously developed both for pre-service and in-service such as follows:

Pre-Service Teachers Training:

- **Enriched Seminars**: Pre-service seminars should be designed to equip teachers with a deep understanding of informatics concepts and pedagogical approaches. This includes exploring real-world applications, integrating technology into the curriculum, and fostering a growth mindset.

- **Focus on Authentic Learning**: Provide current teachers with hands-on workshops and training modules that integrate real-world problems and scenarios into their teaching of informatics. This could include the use of project-based learning, problem-solving tasks, and collaborative learning environments that mirror industry practices.

- **Gender-Inclusive Practices**: Offer strategies and tools to promote gender equity in the classroom, such as inclusive language, diverse role models, and gender-sensitive teaching methods that encourage all students, regardless of gender, to engage with technology and informatics. This can involve awareness-building around unconscious biases and promoting an inclusive learning environment.

In-Service Teachers Professional Development:

- **Informatics Areas & Competencies**: Teachers, especially those who are generalists or teaching informatics as integrated into other subjects, should be equipped with a robust understanding of both core informatics concepts and effective pedagogical approaches. This includes providing training on fundamental informatics concepts (e.g. programming, data, etc.) as well as guidance on effective teaching methods such as project-based learning. Additionally, training should support teachers in aligning informatics education with broader educational goals and standards, and in integrating technology tools and resources into their instruction.
- **Authentic Learning Environments**: In-service training should focus on strategies for creating authentic learning experiences that connect informatics to real-world problems. This includes incorporating project-based learning, problem-solving, and collaborative activities.
- **Gender-Inclusive Practices**: Teachers should receive training on gender-inclusive pedagogy to ensure that all students feel valued and have equal opportunities to participate in informatics education. This involves addressing stereotypes, conscious and unconscious gender biases of teachers, understanding structural inequalities relating to gender, especially within school systems, promoting diversity, and creating a supportive learning environment for all students.

By fostering a well-defined authentic context through the authentic learning principles previously discussed (e.g. project-based learning, real-world applications, and collaborative activities), a more inclusive and equitable learning environment for all students could be created. This approach can help to break down gender stereotypes, encourage diverse participation, and increase the chances of all people, regardless of gender, succeeding in the field of informatics. Therefore, by addressing these aspects, this pillar aims to empower teachers to create engaging, relevant, and inclusive learning environments that meet the diverse needs of all students.

2.5. Building an Interconnected Framework

The framework's effectiveness hinges on the interconnection of the four pillars, each reinforcing and supporting the others. All key areas of the framework overlap, demonstrating the interconnectedness of informatics, authentic learning, gender inclusion, and teacher professional learning. Teachers need to be equipped with the knowledge and skills to create inclusive learning environments while also understanding how to integrate informatics into authentic learning experiences. This requires a multifaceted approach that addresses both pedagogical and content-specific needs. Teaching informatics in an authentic way, which can also include gender-inclusive practices, helps students understand the relevance of the subject to their lives and careers, fostering greater engagement and motivation. Therefore, to ensure the effective implementation of the informatics framework, it is important to consider the interconnectedness of its constituent pillars through a systemic approach. For example:

- **Informatics Areas and Competencies (A, D):** This area connects Pillar A (Informatics Areas & Competencies) and Pillar D (Teacher Professional Learning), suggesting that teachers need to be equipped with the knowledge and skills to analyse curricula from an informatics perspective.
- **Gender Inclusion (C, D):** This area links Pillar C (Gender Inclusion) and Pillar D (Teacher Professional Learning), highlighting the importance of preparing teachers to create inclusive learning environments.
- **Authentic Learning (A, B):** This overlap between Pillar A (Informatics Areas & Competencies) and Pillar B (Authentic Learning) suggests that informatics can be used to enhance authentic learning experiences.
- **Teacher Professional Learning (B, C, D):** This area connects all three pillars, indicating that teacher practices and development are crucial for integrating informatics, promoting authentic learning, and fostering gender-inclusive classrooms.

- **Continuous Monitoring and Evaluation:** Regular assessment of all aspects of the framework is essential to measure progress, identify areas for improvement, and ensure ongoing effectiveness.

3. The TINKER Toolkit

The TINKER framework provides a structure for creating engaging and effective learning experiences in informatics education. It consists of four key elements – Informatics Areas & Competencies, Authentic Learning, Gender Inclusion, and Teacher Professional Learning – that guide teachers in designing age-appropriate lessons. The final proposed framework forms the basis for designing and developing the TINKER Toolkit, which includes age-appropriate learning scenarios for teaching and assessing informatics competencies. In this way, teachers, as the primary target group, will be equipped with the necessary skills to apply this pedagogical framework and design learning scenarios for classroom use.

The framework also serves as a foundation and reference point beyond the project's duration. Based on it, the TINKER Toolkit is developed to show how to apply the framework when preparing learning scenarios for teaching informatics in upper primary and lower secondary education. It acts as a practical guide for teachers, offering guidelines for designing scenarios in line with national curricula and templates for reflecting on their teaching practice.

More specifically, the TINKER Toolkit includes the following information:

1. **Guidelines** to design learning scenarios and activities based on the TINKER framework in line with national curricula.
2. **Template** for teachers to design learning scenarios using the Framework.
3. A **self-reflective tool** (aligned with SELFIE) for teachers to reflect on their teaching practice - whether it follows the TINKER framework (authentic learning and gender-inclusive).
4. A **collection of 100 learning scenarios** for upper primary and lower secondary education (50 per education level).

3.1. Guidelines to design learning scenarios and activities based on the TINKER Framework

For the development of the learning scenarios, TINKER uses the Framework developed based on the core areas of informatics proposed by Informatics4All (e.g. data and information, algorithms, programming), enhancing and framing it with the principles of authentic learning and gender inclusion. Table 2 summarises the key elements of each pillar of the TINKER Framework.

Table 2. TINKER Framework Pillars

Pillar A. Informatics Areas and Competencies	**Pillar B. Authentic Learning**	**Pillar C. Gender Inclusion**	**Pillar D. Teacher Professional Learning**
1. Data and Information 2. Algorithms 3. Programming 4. Computing Systems 5. Networks and Communication 6. Human – Computer Interaction 7. Design and Development 8. Digital Creativity 9. Modelling and Simulation 10. Privacy, safety, security 11. Responsibility and empowerment	1. Authentic Context 2. Authentic Task 3. Expert Performance 4. Multiple Perspectives 5. Collaboration 6. Scaffolding 7. Articulation 8. Reflection 9. Authentic Assessment	1. More Diverse Role Models in STEM 2. Gender-Inclusive Learning Materials & Language 3. Equitable Engagement in Collaborative Learning Environments 4. Inclusive Classroom Interactions 5. Real-World Applications of STEM 6. Encouraging Research and Real-world Applications of IT beyond the Classroom 7. Normalising Failure and Encouraging Perseverance 8. Diverse Teaching Strategies	1. Training on Informatics Areas and Competencies, Authentic Learning and Gender-Inclusive Methodologies 2. Resources for Effective Teaching Practices 3. Support for Ongoing Professional Development

The three main pillars (A, B, C) are addressed along with specific steps and guidelines to integrate the framework elements into learning scenario design and development.

Step 1: Curriculum Analysis

Begin by **reviewing the national curriculum** to ensure alignment with educational standards and learning goals. **Identify at least one** learning scenario **for each of the 11 Informatics Areas** (Table 2) defined in the TINKER Framework. This step ensures that the selected scenarios cover a broad spectrum of informatics topics, giving students a well-rounded understanding of essential competencies. Consider the specific requirements and learning outcomes for each area to develop relevant, engaging scenarios.

Step 2: Set the Learning Objectives

Clearly **define the learning objectives for each scenario**. These should specify what students will be able to understand or accomplish by the end of the lesson. Make sure the objectives are measurable and tied to both the national curriculum and the TINKER Framework. Well-defined objectives will guide the scenario's structure and ensure that students are working towards concrete, assessable skills in informatics.

Step 3: Prepare the Scenario Setting and Activities' Content

Design authentic learning activities that provide real-world context and make learning relevant to students' lives. The activities should last between **20 and 45 minutes** to maintain engagement while fitting within typical class times. Focus on creating gender-inclusive activities that encourage participation from all students, ensuring equal access to learning opportunities. Incorporate diverse perspectives and remove potential barriers, fostering an inclusive environment that supports the learning needs of all students, regardless of gender.

How to design authentic learning activities?

Authentic learning activities help students engage in real-world tasks and develop practical skills. In the field of informatics, designing these activities fosters problem-solving, collaboration, and critical thinking, as well as deepening understanding of technology's role in daily life. Below are guidelines for primary and secondary school teachers to design effective, engaging and meaningful informatics authentic learning activities that help students connect classroom learning to real-world applications.

- **Create an Authentic Context**
    - Use real-world environments where knowledge and skills are applied as they would be in everyday life.
    - Avoid oversimplifying tasks to ensure students are motivated to engage in meaningful learning.
- **Develop Authentic Tasks**
    - Assign open-ended, interdisciplinary tasks that reflect real-world problems.
    - Tasks should require sustained investigation and focus on production rather than reproduction of knowledge.
- **Incorporate Expert Performance**
    - Provide students access to experts and role models to observe real-life applications.

    - Invite experts to share their experiences, allowing students to understand professional thought processes and problem-solving strategies.
- **Offer Multiple Perspectives**
    - Encourage students to view problems from various roles and stakeholders' perspectives.
    - Have students engage with different users or clients to identify diverse needs and design appropriate solutions.
- **Foster Collaboration**
    - Design tasks that require teamwork, dividing roles and responsibilities within groups.
    - Promote collaborative success, ensuring that all students contribute to achieving a shared goal.
- **Encourage Articulation**
    - Provide opportunities for students to explain and present their ideas and solutions.
    - Encourage public presentations and discussions to foster clarity of thought and improve communication skills.
- **Support Reflection**
    - Build in time for students to reflect on their decisions, both during and after the learning process.
    - Use reflective journals and reports to help students consider challenges and improvements.
- **Use Scaffolding**
    - Provide initial guidance and support through templates, examples, and coaching.
    - Gradually reduce support as students become more comfortable, encouraging independent problem-solving.
- **Include Authentic Assessment**
    - Integrate assessment within the learning process, focusing on both the final product and the process.
    - Use real-world evaluation criteria, including functionality, collaboration, and the ability to communicate solutions to non-expert audiences.

How to design gender-inclusive activities?

Creating gender-inclusive activities in informatics ensures that all students feel valued, engaged, and capable, regardless of gender. It's crucial to challenge stereotypes and encourage diverse participation. The following guidelines will help primary and secondary school teachers design activities that foster a welcoming environment for all students, empowering them to develop informatics skills and creativity.

- **Introduce More Diverse Role Models in STEM:** Introducing non-binary, and female role models and relatable figures who have succeeded in STEM fields into the curriculum can significantly impact all students' perspectives on their ability to succeed in these areas. Having relatable figures to look up to can encourage more students regardless of their gender identity to pursue STEM careers and believe in their potential.
- **Incorporate Gender-Inclusive Learning Materials & Language:** To make STEM more inclusive, educational materials should avoid perpetuating stereotypes and instead present content in a gender-inclusive way and through a gender-inclusive language, for example through the use of "they" instead of "he/she". This encourages boys and girls and also those who identify as non-binary to engage with STEM subjects without feeling pigeonholed into traditional gender roles.
- **Promote Equitable Engagement in Collaborative Learning Environments:** Collaborative and inclusive learning environments where every student's contribution is valued can help make STEM subjects more appealing to all students. Group projects and teamwork should be structured in a way that promotes equality and respects diverse perspectives.
- **Promote Inclusive Classroom Interactions:** Foster an inclusive classroom environment by valuing all students equally, challenging gender stereotypes, and promoting positive reinforcement. This involves maintaining positive nonverbal communication, addressing peer pressure and bullying, and creating a safe space for all students to learn and succeed.
- **Integrate Real-World Applications of STEM:** By showing how STEM is used to solve real-world problems, including those that disproportionately affect women, the curriculum can become more relevant and engaging. Highlighting the societal impact of STEM can draw more women into these fields.

- **Encourage Research and Real-World Applications beyond the Classroom**: By encouraging research and real-world applications of IT beyond the classroom, all students can be empowered to explore further studies and careers in tech fields.
- **Normalise Failure and Encourage Perseverance:** STEM subjects often involve trial and error, and it's crucial that all students understand that failure is a part of the learning process. Encouraging a growth mindset and resilience in the face of challenges can keep all students engaged in STEM fields.
- **Adapt Teaching Strategies:** Adapting teaching methods to cater to diverse learning styles can help make STEM subjects more accessible to all students, including girls. This might involve using a variety of teaching materials, interactive lessons, and practical applications that appeal to different students.

Step 4: Indicate How You Integrated the TINKER Framework in the Learning Scenario

In this step, reflect on how the TINKER Framework was applied to your scenario:

- **Why is it authentic?** Explain how the activities mirror real-world tasks, allowing students to apply their knowledge in meaningful contexts, based on the TINKER Framework Principles in Pillar B (Authentic Learning).
- **Why is it gender-inclusive?** Describe how the scenario addresses gender inclusivity based on the TINKER Framework Principles in Pillar C (Gender Inclusion), making sure that all students can participate equally. Consider if the materials, examples, and activities avoid reinforcing stereotypes and provide diverse role models.
- **How can level progression be achieved?** Outline how the scenario allows for progression in skill levels, enabling students to build on prior knowledge and move from basic to more complex tasks. This helps cater to different learning paces and ensures continuous development.

In the next section, we present a template and specific instructions on how to design learning scenarios and activities specifically following the above process.

3.2. Template to design learning scenarios and activities based on the TINKER framework

This section presents a template to help teachers design learning scenarios and activities based on the TINKER Framework. Each element of the template is accompanied by a brief description to guide teachers on what information to include.

We also provide a complete example of how the TINKER Framework, aligned with the Informatics4ALL core principles, can be applied to teaching basic programming skills in lower secondary education in Cyprus. This structured approach to integrating framework elements into scenario design equips teachers to support the development of key informatics skills in their students.

In order to properly develop each learning scenario, teachers should take into consideration the national curriculum of their country and adapt it accordingly.

Table 3. Template of learning scenario development based on the TINKER framework

Learning Scenario Information	
Title	Provide a **short title** (up to 50 characters) that reflects the main focus of the learning activity.
Age Level	Specify the student's **grade level** the activity is intended for.
Duration	Indicate the estimated **duration** of the activity, e.g., 45 minutes
Informatics topic areas	Indicate the **core areas of informatics** that the activity relates to, based on the TINKER and Informatics4All frameworks (e.g., "Algorithms," "Programming," "Simulation" etc.)
Content domain (Integrated Subjects)	Indicate the **school subjects** that the learning activity can integrate with, according to your national curriculum (e.g., mathematics, science, technology etc.)
Learning Objectives	**List 2-4 Objectives of the Learning Scenario**. Start with a verb. Describe specifically what the students will gain by participating in the scenario activity. Use Bloom's Taxonomy to write the Learning Objectives. It's better to focus on what the students should be able to do, not just know. E.g. *Upon completing this activity, the students should be able to:* ● Apply this method … ● Contrast this with that … ● Give examples about that …
Scenario Description	
Setting	**Set the scenario by creating a story**. The scenario should always be linked to at least one learning objective, **focusing on real-life issues** teachers might encounter in their work practices. ● Provide some context for the story (where, who, what's the problem) ○ Refer to the teacher directly (2nd person singular, you). ○ The setting should always ask the teacher 'What should you do?' to indicate that they take action.

<table>
<tr><td>(Digital) Tools</td><td>List all instructional materials and tools (online and offline) needed for the lesson (e.g. books, guides, etc.). Be specific and include both physical items and digital tools.</td></tr>
<tr><td>Activity</td><td>Prepare at least one activity of 20-45 minutes duration.
Provide the teachers with the specific steps required, highlighting the processes they should follow to successfully implement this lesson activity.
The activity's steps should always be aligned with the scenario learning objectives (at least one learning objective should be addressed).
Stick only to must-know information (information necessary for the teachers to conduct the lesson).
Include both plugged and unplugged activities:
Plugged Activities: These will involve the use of digital technologies, such as Bee Bot, Lego WeDo 2.0, Minecraft, or other relevant tools.
Unplugged Activities: Incorporate games, challenges, stories, kinaesthetic engagement, and artwork as core components to facilitate learning without the use of technology.
Create short sentences and paragraphs which are readable without hassle.
Don't use academic writing.</td></tr>
<tr><td>Teachers
and students' Roles</td><td>Define the roles of both teachers and students during the activity. Specify how teachers will guide and support students, and what responsibilities students will have. For instance:
Teachers: "Facilitate discussions and provide support during activities."
Students: "Engage in group work and present findings."</td></tr>
<tr><td>Evaluation/
Assessment</td><td>Describe how you will assess student learning throughout the activity. This can include various methods such as observations of participation and collaboration, quizzes, presentations or rubrics assessing specific criteria.</td></tr>
<tr><td colspan="2">TINKER Framework Integration</td></tr>
<tr><td>How is the activity authentic learning?</td><td>Explain how the scenario connects to real-world applications. Refer to the principles of authentic learning outlined in the TINKER framework and explain how each principle is integrated into the learning scenario.</td></tr>
<tr><td>How is gender inclusiveness ensured?</td><td>Describe how the activity promotes equality and encourages all students to participate actively, avoiding stereotypes or biases in roles.</td></tr>
<tr><td>Considerations for level progression</td><td>Provide suggestions on how to adapt the activity to different skill levels.</td></tr>
</table>

3.2.1. Example of a Learning Scenario

Table 4. Example of a Learning Scenario

Learning Scenario Information	
Title	Designing an Algorithm to Navigate a Physical Maze
Age Level	10-12 years old
Duration	45 minutes
Informatics topic areas	Algorithms
Content domain (Integrated Subjects)	Mathematics, Technology
Learning Objectives	Upon completing this activity, the students should be able to: ● Design a step-by-step algorithm to navigate a maze in a physical space. ● Apply loops and conditionals in their algorithms to solve the maze efficiently. ● Evaluate and debug their algorithm based on its performance during testing.
Scenario Description	
Setting	You are excited to teach your students about algorithms this year. However, many of them find the concept abstract and challenging to connect to their daily lives. You want to engage them in a hands-on activity where they design algorithms without the use of computers, allowing them to see how algorithms apply to everyday tasks.
(Digital) Tools	● Computer/Laptop ● Projector/ Whiteboard ● Markers ● Classroom with space cleared for a physical maze, such as desks/chairs or tape on the floor, forming a simple maze with obstacles.

<table>
<tr>
<td>Activity</td>
<td>
Step 1 (10 minutes): Introduction to Algorithms Using Real-Life Examples

Explain to the students what an algorithm is with the following real-world example.
Creating a sandwich is a classic example of an algorithm. Present students with the video “Program your teacher to make a Jam Sandwich.” (3’44’’) and discuss. Then, ask them to explain the process step by step and write the steps on the interactive whiteboard. Here’s a simple step-by-step breakdown:

Gather Ingredients: Bread, fillings (like ham, cheese, or vegetables), and condiments (like mustard or mayonnaise).
Lay Out Bread: Place two slices of bread on a clean surface.
Add Condiments: Spread condiments on one or both slices of bread.
Layer Fillings: Add your chosen fillings evenly on one slice.
Top with Second Slice: Place the second slice of bread on top of the fillings.

Explain that each step must be followed in sequence to achieve the desired outcome and provide the below definition:

An algorithm is a systematic, step-by-step procedure or set of rules designed to perform a specific task or solve a problem. Algorithms are not limited to computer science; they are present in many everyday activities, guiding us through routine tasks efficiently and effectively.
Recipes are structured algorithms that guide you through cooking. For example, baking cookies involves specific steps like mixing ingredients, shaping dough, and baking at a certain temperature for a defined time.
Step 2 (10 minutes): Understanding Maze Navigation

Show a simple map of a maze on the whiteboard. Explain that just like following the sandwich-making steps, solving a maze requires a clear sequence of instructions, or an algorithm, to navigate from start to finish.
Introduce the terms loops and conditionals:

Loops: Explain that loops are used when a set of steps needs to be repeated (e.g., “Move forward until you reach a wall”).

</td>
</tr>
</table>

	○ **Conditionals**: Discuss how conditionals help the algorithm make decisions based on specific conditions (e.g., "If you encounter a wall, turn left"). ● Walk the students through the basic maze on the whiteboard. Have them suggest directions (e.g., "Turn right", "Move forward three steps"), and discuss how this resembles programming a robot to follow commands. **Step 3 (20 minutes): Designing and Testing Algorithms in a Physical Maze** ● Arrange the chairs and desks or tape on the floor to form a simple maze in the classroom. The maze should have a start point, an endpoint, and several obstacles or turns to make the path challenging but achievable within the time limit. ● Divide students into mixed-gender pairs or small groups. One student or more students in each pair will act as the "programmer(s)", and one student will be the "robot". The "robot" will be blindfolded to simulate the idea that they must rely solely on the algorithm to navigate the maze. ● The "programmer(s)" will create a step-by-step set of instructions (an algorithm) to guide the blindfolded "robot" through the maze. These instructions must be specific and clear (e.g., "Take three steps forward, turn right, move forward until you reach the wall"). ● Once the algorithm is written, the "robot" will follow the programmers' instructions, moving through the maze step by step. The "robot" must only move based on what the programmer says, and the programmer(s) must communicate their instructions clearly. ● If the robot encounters an obstacle or follows the instructions incorrectly, the team will stop, evaluate the mistake, and the programmer(s) will revise the algorithm (debugging) to fix any errors. ● If the "robot" makes an incorrect move (e.g., bumping into a wall or missing a turn), the programmer must identify where the mistake occurred and adjust the instructions. This simulates real-world debugging, where errors in code must be found and corrected for a programme to run correctly.

	• Throughout the activity, give structured questions on key issues they need to consider. As students become more comfortable, gradually, the help is removed. Some examples of questions are: ○ *What steps are you giving the robot to help it move through the maze?* ○ *What went wrong when the robot hit an obstacle? How can you fix it?* ○ *How can you make your instructions easier for the robot to follow?* **Step 4 (5 minutes): Group Reflection and Discussion** • Bring the class together to reflect on the activity. Ask each group to share and compare their algorithms, and describe the challenges they faced while guiding their "robot" through the maze. Encourage them to talk about how they used loops or conditionals to improve the efficiency of their instructions. • **Prompts for Discussion:** ○ *"What steps in your algorithm worked best to guide your robot through the maze?"* ○ *"How did you use loops to repeat actions, and did it make your algorithm more efficient?"* ○ *"What conditionals did you use to handle obstacles? Did they work as expected?"* ○ *"How did you improve or debug your algorithm when the robot made a mistake?"*
Teachers and students' Roles	**Teachers**: Guide the students through the initial examples and assist them in understanding algorithms, loops, and conditionals. Supervise the maze activity. While working, pose additional reflective questions to reveal their thinking and push them to progress, finding their solutions when challenges arise. Encourage students to reflect on errors and adjustments. **Students**: Act as programmers and robots. As programmers, they create and test algorithms, and as robots, they follow the instructions exactly to highlight the importance of precise commands.

Evaluation/ Assessment	• Evaluate each group's ability to design a clear, functional algorithm that successfully navigates the maze. Assess their use of loops and conditionals, as well as their debugging strategies. • Observe student participation during the maze activity, paying attention to how they create and refine their algorithms.
TINKER Framework Integration	
How is the activity authentic learning?	The activity uses real-world problem-solving through hands-on activities. Students design and test algorithms in a tangible environment, making the abstract concept of algorithm design more concrete. Almost all principles of authentic learning are applied in this activity setting including authentic context and tasks, multiple perspectives, collaboration, reflection, scaffolding and authentic assessment.
How is gender inclusiveness ensured?	The activity encourages collaboration in mixed-gender teams, promotes equal participation, and avoids gendered roles, ensuring all students contribute as both programmers and robots.
Considerations for level progression	For younger or less experienced students, simplify the maze and focus more on the basic structure of step-by-step instructions. For older or more advanced students, more complex mazes or additional algorithmic concepts such as functions or nested conditionals can be introduced to deepen understanding.

3.3. The TINKER Self-Reflective Tool

The self-reflective tool below is designed to assist teachers in evaluating their teaching practices against the TINKER framework. The tool is aligned with the philosophy of the European Commission's SELFIE tool and, in particular, the area of Pedagogy included in it. During self-reflection, there are two main aspects to consider: mark whether your teaching practice follows the principles of authentic learning and gender inclusion.

Table 5. Self-Reflective Tool

Authentic Learning Principles			
Principle	**Yes**	**No**	**Comments**
Authentic contexts. There is a virtual or physical environment reflecting how knowledge is used in real life.			
Authentic tasks and **activities.** There are complex tasks without predefined steps students have to follow. These tasks are relevant to the real-world and interdisciplinary. They require production (not reproduction) and cannot be solved on the spot (sustained investigation over a period).			
Expert performances and modelling. There is access to expertise, seeing how experts think and work, observing real-life episodes, and having opportunities to share stories.			
Multiple roles and **perspectives**. There is an opportunity to adopt various roles and see things from different points of view.			

Authentic Learning Principles			
Principle	**Yes**	**No**	**Comments**
Collaborative construction of knowledge. Tasks are addressed to groups, for individuals to work in pairs or teams, striving for whole-team success.			
Articulation to enable tacit knowledge to be made explicit. There is an opportunity to articulate thoughts and results, present an argument publicly, and reach an understanding through social interaction.			
Reflection. There is an opportunity to think about, reflect and discuss choices in action (while making choices) or after action (after decision-making).			
Coaching and **scaffolding.** There is assistance, coaching, and guidance from the teacher at critical times, which activate metacognition.			
Authentic assessment. The assessment is integrated within the learning activity, rather than being a separate function, where students use a wide range of skills, showcasing performance or products to be evaluated with proper criteria (aligned with the task).			

Gender Inclusion Principles			
Principle	**Yes**	**No**	**Comments**
More diverse role models in STEM. **Non-binary**, and **female** role models who have succeeded in STEM fields are introduced into the curriculum.			
Gender-inclusive learning materials & language. Educational materials avoid perpetuating stereotypes and present content in a gender-inclusive way and through gender-inclusive language. For example, use of "they" instead of "he/she".			
Equitable engagement in **collaborative** learning environments**.** Tasks are being addressed to groups in ways that every student's contribution is valued.			
Inclusive classroom **interactions.** All students feel valued and supported in the classroom environment through equal attention and positive reinforcement to all students.			
Real-world applications of STEM. Real-world problems are integrated into tasks or activities to improve people's lives.			

Gender Inclusion Principles			
Principle	Yes	No	Comments
Encourage research and **real-world applications** of **IT beyond the classroom.** Additional resources are provided and successful diverse role models in IT are highlighted to empower all students to explore further studies and careers in tech fields.			
Normalising failure and **encouraging perseverance.** A growth mindset and resilience are encouraged in the face of challenges to understand that failure is a part of the learning process.			
Diverse teaching strategies. Different teaching methods are adapted to cater to diverse learning styles. This might involve using various teaching materials, interactive lessons, and practical applications that appeal to different students.			

3.4. Collection of 108 learning scenarios for upper primary and lower secondary education

In line with the proposed framework and with consultations with teachers, we have developed a Toolkit with age-appropriate learning scenarios for teaching and assessing informatics competencies in upper primary and lower secondary education. The Toolkit is designed to be adaptable. Teachers can effectively align the scenarios with their national curriculum and student needs by evaluating their own learning objectives and adapting the scenarios accordingly. For access to the full set of the age-appropriate learning scenarios, please visit the Tinker project website here: https://tinker-project.eu/resources/framework-and-toolkit/.

4. Conclusion

The TINKER framework emerged from a robust development process, building upon the three initial pillars outlined in the project proposal: informatics topics and competencies, authentic learning, and gender inclusion. Extensive desk and field research in the partner countries validated these pillars as essential elements for effective informatics education. This research, combined with practical considerations, led to the identification of an additional crucial pillar: Teacher Professional Learning.

The TINKER framework facilitates the design and development of engaging learning scenarios by providing a structured approach that integrates these core pillars:

- Informatics Areas & Competencies: The framework ensures learning scenarios align with essential informatics concepts and skills, preparing students for future challenges.
- Authentic Learning: By emphasising real-world applications, the framework sparks students' interest and encourages them to think critically about solving relevant problems.
- Gender Inclusion: The framework promotes inclusivity through gender-neutral language and by highlighting more diverse role models, fostering a welcoming environment for all students.
- Teacher Professional Learning: Recognising the importance of well-equipped teachers, the framework emphasises providing teachers with the pedagogical skills necessary to facilitate engaging learning experiences.

The TINKER framework empowers teachers to create dynamic learning scenarios by guiding them through five steps:

- Curriculum Analysis: Aligning learning scenarios with existing informatics curriculum ensures a holistic learning experience.
- Teacher Practices and Development: Utilising resources and training opportunities equip teachers to teach in an authentic and gender-inclusive approach.
- Authentic Learning Activities: Designing activities that connect informatics concepts to real-world situations fosters deeper understanding and engagement.
- Gender-inclusive Activities: Designing activities that are inclusive and avoid perpetuating gender stereotypes. These activities should be inclusive and avoid perpetuating gender stereotypes. For example, using a bakery scenario doesn't have to assume ownership or leadership roles are inherently male.

- Monitoring and Evaluation: Regularly assessing the effectiveness of learning scenarios allows for continuous improvement and ensures the framework remains a valuable tool for teachers. This evaluation process should consider inclusivity and assess if the learning environment is welcoming for all students, regardless of gender.

A key strength of the TINKER framework is its flexibility. The core pillars can be adapted to diverse educational needs and contexts, ensuring its applicability in various learning environments.

5. References

Allen, C.D. & Eisenhart, M, (2017) Fighting for Desired Versions of a Future Self: How Young Women Negotiated STEM-Related Identities in the Discursive Landscape of Educational Opportunity. *Journal of the Learning Sciences*, *26*(3), 407-436. https://doi.org/10.1080/10508406.2017.1294985

Brett, L. (2022, March 2024). *Women in STEM in the European Union – Facts and Figures*. European Student Think Tank, Retrieved December 17, 2024, from https://esthinktank.com/2022/03/24/women-in-stem-in-the-european-union-facts-and-figures/

Chan, R.C.H. (2022). A social cognitive perspective on gender disparities in self-efficacy, interest, and aspirations in science, technology, engineering, and mathematics (STEM): the influence of cultural and gender norms. *International Journal of STEM Education*, *9*, 37. https://doi.org/10.1186/s40594-022-00352-0

Christou, E., Parmaxi, A., Perifanou, M., & Economides, A. A. (2022, June). Gender-Sensitive Materials and Tools: The Development of a Gender-Sensitive Toolbox Through National Stakeholder Consultations. In *International Conference on Human-Computer Interaction* (pp. 485-502). Springer International Publishing.

Christou, E., Parmaxi, A., Perifanou, M., & Economides, A.A. (2022). *Gender-Sensitive Materials and Tools: The Development of a Gender-Sensitive Toolbox Through National Stakeholder Consultations*. In: Meiselwitz, G. (eds) *Social Computing and Social Media: Design, User Experience and Impact.* Springer. https://doi.org/10.1007/978-3-031-05061-9_34

Cole, N. S. (1990). Conceptions of educational achievement. *Educational researcher*, *19*(3), 2-7.

Demirkol, K., Kartal, B., & Tasdemir, A. (2022). The Effect of Teachers' Attitudes towards and Self-Efficacy Beliefs Regarding STEM Education on Students' STEM Career Interests. *Journal of Science Learning*, *5*(2), 204-215.

DeWitt, J. & Archer, L. (2015). Who Aspires to a Science Career? A comparison of survey responses from primary and secondary school students. *International Journal of Science Education*, *37*(13), 2170-2192. https://doi.org/10.1080/09500693.2015.1071899

De Wit, S., Hermans, F., Specht, M., & Aivaloglou, E. (2023). Children's Interest in a CS Career: Exploring Age, Gender, Computer Interests, Programming Experience and Stereotypes. In *Proceedings of the 2023 ACM Conference on International Computing Education Research (ICER 2023)* (Vol. 1, pp. 245-255). Association for Computing Machinery (ACM). https://doi.org/10.1145/3568813.3600131

Eagly, A. H. (2021). Hidden in plain sight: The inconsistent gender gaps in STEM and leadership. *Psychological Inquiry*, *32*(2), 89–95.

European Commission. (2024). ICT specialists in employment. Eurostat. Retrieved December 17, 2024, from https://ec.europa.eu/eurostat/statistics-explained/index.php?title=ICT_specialists_in_employment

European Commission, Directorate-General for Communications Networks, Content and Technology, (2019). 2nd survey of schools : ICT in education : objective 1 : benchmark progress in ICT in schools, final report, Publications Office. https://data.europa.eu/doi/10.2759/23401

European Commission, European Education and Culture Executive Agency, (2022). Informatics education at school in Europe, Publications Office of the European Union. https://data.europa.eu/doi/10.2797/268406

Garriott P.O., Hultgren K.M. & Frazier, J. (2017). STEM stereotypes and high school students' math/science career goals. *Journal of Career Assessment, 25*(4), 585–600.

Gee, E., Tran, K. M., & Parekh, P. (2020). Designing analog games that engage girls with computer science concepts. *International Journal of Designs for Learning*, *11*(2), 17–26.

Happe, L., Buhnova, B., Koziolek, A. and Wagner, I. (2021). Effective measures to foster girls' interest in secondary computer science education: A Literature Review. *Education and Information Technologies, 26*, 2811–2829. https://doi.org/10.1007/s10639-020-10379-x

Herrington, J., & Oliver, R. (2000). An instructional design framework for authentic learning environments. *Educational technology research and development*, *48*(3), 23-48.

Herrington, J., Reeves, T. C., & Oliver, R. (2014). *Authentic learning environments* (pp. 401-412). Springer New York.

Kuteesa, K. N., Akpuokwe, C. U., & Udeh, C. A. (2024). Gender equity in education: addressing challenges and promoting opportunities for social empowerment. *International Journal of Applied Research in Social Sciences*, *6*(4), 631-641.

Lave, J. (1988). *The culture of acquisition and the practice of understanding.* IRL report 88-00087, Institute for Research on Learning.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation.* Cambridge University Press.

Lavy, V., & Megalokonomou, R. (2023). The Short- and the Long-Run Impact of Gender-Biassed Teachers, *American Economic Journal: Applied Economics*, forthcoming.

Main, J.B., & Schimpf, C. (2017). The underrepresentation of women in computing fields: a synthesis of literature using a life course perspective. *IEEE Transactions on Education*, *60*(4), 296– 304.

Malazita, J. W., & Resetar, K. (2019). Infrastructures of abstraction: How computer science education produces anti-political subjects. *Digital Creativity*, *30*(4), 300–312.

Muntoni, F., Wagner, J., & Retelsdorf, J. (2021). Beware of stereotypes: Are classmates' stereotypes associated with students' reading outcomes? *Child Development*, *92*(1), 189–204. https://doi.org/10.1111/cdev.13359

Msambwa, M. M., Kangwa D., Cai L., Antony F. (2023) A systematic review of the factors affecting girls' participation in science, technology, engineering, and mathematics subjects. *Computer Applications in Engineering Education*, *32*(2), 10.1002/cae.22707.

Piaget, J. (2013). *The construction of reality in the child*. Routledge.

Ren, X. (2022) Adopting Feminist Pedagogy in Computer Science Education to Empower Underrepresented Populations: A Critical Review. *TechTrends*, *66*, 459–467. https://doi.org/10.1007/s11528-022-00728-7

Resnick, M. and Rosenbaum, E. (2013). Design, Make, Play: Growing the Next Generation of STEM Innovators, chapter Designing for Tinkerability. Taylor & Francis.

Scardamalia, M., & Bereiter, C. (1994). Computer support for knowledge-building communities. *The Journal of the Learning Sciences*, *3*(3), 265-283.

Sharpe, C. T. V., & Rothenberg, J. (2018). Move slow and fix things: Teaching computer science majors to decode discrimination and design diverse futures. *Transformations: The Journal of Inclusive Scholarship and Pedagogy*, *28*(2), 202–209.

Singer, A., Montgomery, G., & Schmoll, S. (2020). How to foster the formation of STEM identity: studying diversity in an authentic learning environment. *International Journal of STEM Education*, *7*, 1-12.

Stein, S. J., Isaacs, G., & Andrews, T. (2004). Incorporating authentic learning experiences within a university course. *Studies in Higher Education*, 29(2), 239-258.

Szláv, A. (2021). Barriers, Role Models, and Diversity– Women in IT. *CentralEuropean Journal of New Technologies in Research, Education and Practice, 3(3)*, pp. 2027. https://doi.org/10.36427/CEJNTREP.3.3.2582

TINKER project (2024). WP2 A Framework and Toolkit for informatics education: Transnational Report. *TINKER*. https://tinker-project.eu/transnational-report-on-state-of-the-art-and-needs/

Vygotsky, L. (1978). Interaction between Learning and Development. In *Mind in Society*. (Trans. M. Cole), 79-91. Cambridge, Harvard University Press.

Webb, L., Allen, M., & Walker, K. (2002). Feminist pedagogy: Identifying basic principles. *Academic Exchange Quarterly, 6*, 67-72. https://www.researchgate.net/publication/225274654_Feminist_pedagogy_Identifying_basic_principles

Women in Tech: https://www.womentech.net/en-at/how-to/how-can-we-make-stem-curriculum-more-inclusive-women

Washington, N., Barnes, T., Payton, J., Dunton, S., Stukes, F., & Peterfreund, A. (2019). RESPECT 2019: Yes, we still need to talk about diversity in computing. *Computing in Science & Engineering*, *21*, 79–83.