Female Computer Scientists Needed: Approaches For Closing The Gender Gap

Corinna Kröhn
STEM Education
Johannes Kepler University
Linz, Austria
corinna.kroehn@jku.at

Iris Groher
Business Informatics
Johannes Kepler University
Linz, Austria
iris.groher@jku.at

Barbara Sabitzer
STEM Education
Johannes Kepler University
Linz, Austria
barbara.sabitzer@jku.at

Lisa Kuka
STEM Education
Johannes Kepler University
Linz, Austria
lisa.kuka@jku.at

***Abstract*—Despite great effort in research and teaching as well as in raising women's quotas in the public sector and in enterprises, a gender gap in computer science can still be observed. The reasons are manifold, including the lack of interest of girls because of existing stereotypes of male nerds, misconceptions in the field of computer science, and concrete differences in students' self-concept and performance. Girls do not choose schools or studies in the field of computer science because working on computers or coding is unattractive to them. Research shows poorer performance, missing interest, and lower self-concept for girls in secondary computing education as well as a high dropout rate of female students in computer science courses at the bachelor's level. Software development exams contribute to those high dropout rates, and women often choose a different subject or even stop their university education altogether. This leads to fewer female employees in the field of computer science, and this situation calls for special measures at several levels. The current paper provides an overview of the initiatives of our university that aim to address several aspects of the gender gap. The programs offered aim to (1) increase interest in computer science and recruit (highly) gifted girls for our talent programs, (2) recruit and support female bachelor's and master's students in computer science, and (3) redesign programming courses and teaching materials to reduce the gender gap in performance and support, especially for female students. This study also describes a "cyber tutoring" program in which highly gifted girls between 13 and 16 years collaborate with and are supervised by female role models in higher positions in STEM. The evaluation results gained in the first year of this case study are promising and suggest that the extension and further development of the program would be advantageous.**

***Index Terms*—gender gap, talents promotion, computer science, STEM**

I. Introduction

Promoting diversity and fostering equality and inclusion have become strategic missions of many universities globally. At the Johannes Kepler University Linz, for example, equality in educational and professional opportunities is strategically promoted with a dedicated Women Promotion Plan[1]. With respect to education, this plan promotes both women's access to academic studies and the active promotion of female students. However, diversity does not solely manifest in gender disparities. Gardenswartz and Rowe [1] present four layers of diversity dimensions (see Figure 1) that need to be considered: *Personality*, *Internal Dimensions* (e.g. age, gender, race), *External Dimensions* (e.g. work experience, educational background), and *Organizational Dimensions* (e.g. work location, functional level). In this paper, we focus on the gender part of the *Internal Dimensions* in computer science.

Fig. 1. Four layers of the diversity model proposed by Gardenswartz and Rowe [1]

Despite a great effort in research and teaching, a gender gap in computer science can still be observed [2], [3]. At the bachelor's level, this gap typically manifests in the poorer performance of female students than male students as well as higher drop-out rates from computer science courses. This corresponds with our own observations from the introductory programming course in the Business Informatics bachelor's program. Research further shows that the gender gap emerges much earlier, as girls do not even choose schools or studies in the field of computer science because working on computers or coding is unattractive to them. Among young girls, computers and computer science are often perceived as nerdy or geeky [4].

In this paper, we present the initiatives of the Johannes Kepler University Linz that aim to address several aspects

[1]Statute of the Johannes Kepler University Linz, Women Promotion Plan, 21.06.2011.

of the gender gap. The programs offered aim to (1) increase interest in computer science and recruit (highly) gifted girls for our talent programs, (2) recruit and support female bachelor's and master's students in computer science, and (3) redesign programming courses and teaching materials to reduce the gender gap in performance and support, especially for female students.

In Section II, we discuss the relevant literature in the area of gender and computer science. Section III presents our initiatives and early results. We conclude our paper in Section IV with an outlook on future work we plan in this area.

II. Literature Review

There has been extensive research over several decades to understand the lack of female computer scientists. Figure 2 shows the proportion of entering students who plan to major in computer science in the United States. When computing peaked in the mid-1980s, women earned about 35% of bachelor's degrees in computer science in the United States, whereas in 2014 only 18% finished their studies [5].

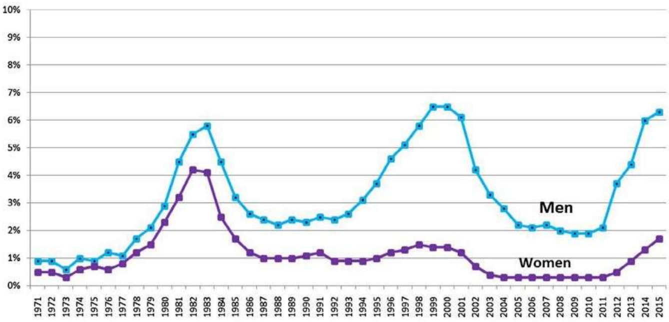

Fig. 2. Proportion of students who plan to major in computer science by gender from 1971 to 2015 in the United States [5]

Hence, women tend to be more attracted to low-paying job fields, which is a significant factor in the gender pay gap. Nonetheless, their diverse perspectives could improve technology development in areas in which companies have overlooked women and their needs (e.g. voice recognition). Many organizations have partnered with industry giants such as Microsoft and Google to help girls find their way into computer science-related areas. However, those efforts have not increased the number of women undertaking computer science majors [5]. Studies show that young women tend to rate their academic abilities, especially their math and computer science abilities, lower than men of the same age, even if they gain similar levels of academic achievement [6] or similar test scores [7]. Indeed, many studies have found a correlation between confidence in computing and/or self-efficacy and the under-representation of women in this field [5].

Concerning Lagesen's analysis, the problem can be categorized into three "deficit models" [8]. The first focuses on women's lower knowledge or experience in the field ("deficits in women"). The second ("deficits in computer science") focuses on improving computer science education to attract women. The third concentrates on the "image" of computer science that lots of people still identify as the typical male stereotype of "hackers," "geeks," and male "geniuses" [8]. Interestingly, this stereotype is much more about racism than it is about gender. When asked, people describe nerds as mostly Asian men devoid of emotion and caring only for money, whereas the majority see black people as unsuited to a career in computer science [9].

Furthermore, the literature concentrates on the lack of female role models in computer science. It seems difficult for girls to come into direct contact with successful women who have majored in male-dominated fields. For example, several studies show the impact of female teachers on female students, particularly for STEM majors [10].

III. Promoting Female Students at our University

In general, we concentrated on three aims to foster female students' interest in computer science and help them during their studies. Concerning the first group, we implemented the "cyber tutoring" program described in Section III-A. By redesigning programming courses on the university's curriculum, we aim to reduce dropout rates, as outlined in Section III-B. In Section III-C, the approaches for supporting female bachelor's students are characterized.

A. *Recruiting gifted girls*

The cyber tutoring program supports gifted girls to allow them to gain basic experience with creative projects in an academic setting. The two-year program started in autumn 2018 and comprised 13 girls between the ages of 13 to 16, who were brought together with women in higher positions in the field of STEM. Several teams were formed after the tutors had proposed their project ideas. The girls not only developed their own projects and products, but were also involved in the research projects of the department, worked after an introduction as peer tutors in the *COOL Lab*, and participated in the new tutor training for basic digital education. The main interaction took place over online platforms such as Moodle, Facetime, Skype, and Dropbox over the course of a year when participants met to present their work-in-progress and forge new collaborations. A vivid feedback culture was fostered to improve and combine projects as well as discuss necessary adaptations. Additionally, accompanying leadership seminars were offered to participants. The girls not only had the opportunity to improve their academic and technical understanding, but also were trained to pass on their know-how on computational thinking to peers at their own school. The program with our first participants will end in July 2020. Hence, we will aim to evaluate the outcome in the upcoming months.

B. *Redesigning programming courses*

Whether at school or university, learning to code is often difficult for beginners. High dropout rates, poor performance, and a major gender gap verify these experiences [11]–[13]. There already exist several approaches to support beginners such as tutoring systems and extra training courses; nonetheless, many students still do not pass their exams. This concurs with our experience of the introductory programming course in the Business Informatics bachelor's program at Johannes Kepler University. We have regularly seen high dropout rates and moderate results by those who passed, and we thus decided to introduce *COOL Programming* into our programming courses [14].

COOL Programming is based on the concept of *COOL Informatics*, a teaching approach that brain-supporting teaching methods and materials [15]. The lessons and tasks are designed based upon the following neurodidactic principles:

1) New content is always built on existing knowledge and learning occurs through association [16], [17].
2) Knowledge cannot be transferred but must be constructed in each student's brain [18].
3) Learning occurs through imitation [16].
4) The brain recognizes and produces patterns, categories, and rules itself [17].
5) The instruction method impacts the retention of new information [16].
6) Double-coded is double-saved (multimedia effect) [18].

COOL integrates four main principles in lecture and task design to make teaching more effective and understandable: discovery, cooperation, individuality and activity.

(1) Discovery learning is supported by solution-based learning, step-by-step instructions and tasks, video tutorials, and observational learning.

(2) Cooperation, a crucial principle of COOL, is supported by team- and groupwork, peer tutoring and peer teaching, and pair programming. *Peer learning* is the use of teaching and learning strategies in which students learn with and from each other without any direct intervention by a teacher. Examples of peer learning are student-led workshops, team projects, study groups, and student-to-student learning partnerships [19]. In contrast to peer learning, *peer teaching* involves advanced students in class acting as tutors [19]. In *pair programming*, two programmers work together on the same algorithm at one computer. One person is the "driver" and the other is the "navigator." The driver types at the computer and performs the physical input, while the navigator observes and comments on the role of the driver [20].

The third principle of COOL is *individuality*, which is supported by self-organized learning with compulsory and optional tasks. Fourth, *activity* is realized by hands-on/minds-on learning by doing, learning by animation, and simulation by playing and designing games (creative learning).

The crucial elements in both lesson and task design in our programming course are the principles of cooperation and discovery as well as individuality (i.e. freedom of choice for students regarding the type, number, and context of exercises). The script and given tasks are designed especially for discovery learning as well as for training certain competencies and programming concepts.

At the beginning of lessons (*Questioning Phase*), students work in small teams supervised by a peer tutor (a student with good programming competencies) and can ask their peers, peer tutors, and docents questions to clarify understanding. The next phase (*Discovering Phase*) is based on the automatic brain function of patterning [21] and is dedicated to extracting the structures, rules, and other essential elements in different examples of Java code. Students use several forms of materials appropriate for discovery learning, such as reading corners (short pieces of correct Java code), sample solutions, step-by-step exercises, and worked examples. The rest of the lecture (*Pair-Programming Phase*) offers the possibility for students (at a similar knowledge level) to work together on their programming tasks, following the rules of pair-programming. Students can ask their partner as well as other peers and tutors when problems with tasks arise.

The outcomes of the first group in 2018 were promising, as the overall motivation as well as learning outcomes improved. In 2019, the experiment was repeated and the course results improved further. The overall success rate of 77% was much higher than that in previous semesters, where it was typically below 50%. We also found that under the COOL approach, students tend to stay on the course longer before they give up. This trend is also visible in homework hand-ins. Interestingly, the gender gap is reduced as well, maybe because women prefer the conversational, cooperative style of the courses to traditional ones. Indeed, while female students increased their performance during the course, male students stayed at the same level. In general, course evaluation was also positive. In particular, students liked the individual feedback and support, small peer groups, and assignments that helped them better understand and learn the different topics.

C. *Supporting female bachelor's students*

There are pragmatic as well as principled reasons for concentrating on peer learning at universities. As both peer learning and peer teaching concentrate on students, universities can easily increase the number of students in classes without more input from staff. Collective forms of learning may better suit women and minority groups than traditional individualistic teaching methods because they concentrate on cooperation rather than competition [19]. Considering the gender gap, more and more of these teaching and learning concepts should thus be investigated.

Although one of the beginners programming classes switched to the concepts of *COOL Programming*, many other courses required support for women struggling to start to code. Consequently, in the winter term of 2018, we created a peer-learning group that met each week in the conference room of our department to discuss upcoming programming tasks [22].

In the summer term 2019, the first official peer-learning class was implemented. Students could take that class voluntarily but still receive ECTS for it. Two PhD students acted as tutors but did not solve the coding tasks step-by-step. Sadly, no female students participated [22].

In the following winter term, the class was restricted to students enrolled in the beginners programming classes of the Computer Science and Business Informatics bachelor's programs. The system also changed from peer learning to peer teaching, as two advanced students were members of the same group. In total, 18 students attended class of which eight were women. Seven of those eight female students passed the final exam [22].

IV. Conclusion and Future Work

To lower the gender gap in the field of computer science, it seems important to introduce strategies to increase the interest of female students, not only at the university level, but also in earlier stages. At the Johannes Kepler University Linz, three strategies have been implemented to promote female students. First, recruiting gifted girls via cyber tutoring can bring young women together with female leaders in a technical field. Female role models can influence young female students, especially in STEM. Second, redesigning programming courses by implementing modern teaching and learning methods such as *COOL Programming* can improve learning outcomes and lower the gender gap. Third, supporting female bachelor's students through peer-learning programs fosters a cooperative and conversational learning style.

Although our strategies have shown an immediate effect on the motivation and support of already enrolled and highly gifted students, it is also vital to foster a curiosity about sciences from an early age by providing appealing, creative, low-threshold projects. In the future, we plan to focus on providing explicit support for students with diverse educational backgrounds, ages, and migration statuses.

References

[1] L. Gardenswartz and A. Rowe, *Diverse Teams at Work: Capitalizing on the Power of Diversity*. Society for Human Resource Management, 2003. [Online]. Available: https://books.google.at/books?id=8UXyAAAAMAAJ

[2] G. G. Gap, "World economic forum," *Cologny/Geneva*, 2017.

[3] S. W. Han, "National education systems and gender gaps in stem occupational expectations," *International Journal of Educational Development*, vol. 49, pp. 175–187, 2016.

[4] B. Rasmussen and T. Håpnes, "Gendering technology: Young girls negotiating ICT and gender," *Merete Lie (ed.), He, She and IT – Revisited (Oslo: Gyldendal Akademiske)*, pp. 173–197, 2003.

[5] K. J. Lehman, L. J. Sax, and H. B. Zimmerman, "Women planning to major in computer science: Who are they and what makes them unique?" *Computer Science Education*, vol. 26, no. 4, pp. 277–298, Dec. 2016. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/08993408.2016.1271536

[6] L. J. Sax, *The gender gap in college: maximizing the developmental potential of women and men*. John Wiley & Sons, 2015.

[7] L. J. Sax, M. A. Kanny, T. A. Riggers-Piehl, H. Whang, and L. N. Paulson, "But I'm not good at math: The changing salience of mathematical self-concept in shaping women's and men's stem aspirations," *Research in Higher Education*, vol. 56, no. 8, pp. 813–842, Dec. 2015.

[8] V. A. Lagesen, "Getting women into computer science," in *Technologies of Inclusion. Gender in the Information Society*, K. H. Sorensen, W. Faulkner, and E. Rommes, Eds. Trondheim: Tapir Academic Press, 2011.

[9] D. Michell, A. Szorenyi, K. Falkner, and C. Szabo, "Journal of Higher Education Policy and Management Broadening participation not border protection: how universities can support women in computer science," *Journal of higher education policy and management*, 2017. [Online]. Available: https://www.tandfonline.com/action/journalInformation?journalCode=cjhe20

[10] C. Porter and D. Serra, "Gender differences in the choice of major: The importance of female role models," Southern Methodist University, Department of Economics, Departmental Working Papers 1705, Dec. 2017. [Online]. Available: https://ideas.repec.org/p/smu/ecowpa/1705.html

[11] M. Guzdial and E. Soloway, "Teaching the Nintendo generation to program," *Commun. ACM*, vol. 45, no. 4, pp. 17–21, Apr. 2002. [Online]. Available: http://doi.acm.org/10.1145/505248.505261

[12] E. Lahtinen, K. Ala-Mutka, and H.-M. Järvinen, "A study of the difficulties of novice programmers," *Acm Sigcse Bulletin*, vol. 37, no. 3, pp. 14–18, 2005.

[13] P. Mozelius, "The gap between generation y and lifelong learners in programming courses–how to bridge between different learning styles," *Open Learning Generations: Closing the Gap from Generation Y to the Mature Lifelong Learners. EDEN 2012*, 2012.

[14] B. Sabitzer, I. Groher, J. Sametinger, and H. Demarle-Meusel, "Cool programming – improving introductory programming education through cooperative open learning," *ICEIT 2020, Oxford, UK*, 2020.

[15] B. Sabitzer, S. Pasterk, and S. Elsenbaumer, "Brain-based teaching in computer science: Neurodidactical proposals for effective teaching," in *Proceedings of the 13th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 197–198. [Online]. Available: https://doi.org/10.1145/2526968.2526994

[16] D. A. Sousa, *How The Gifted Brain Learns*. Corwin Press, 2009.

[17] M. Spitzer, *Lernen – Gehirnforschung und die Schule des Lebens*. Springer Spektrum, 2006.

[18] G. Roth, "Warum sind lehren und lernen so schwierig?" *Zeitschrift für Pädagogik*, vol. 50, 01 2004.

[19] D. Boud, R. Cohen, and J. Sampson, "Peer learning and assessment," *Assessment and Evaluation in Higher Education*, vol. 24, no. 4, pp. 413–426, 1999.

[20] K. W. Han, E. K. Lee, and Y. J. Lee, "The Impact of a Peer-Learning Agent Based on Pair Programming in a Programming Course," *IEEE Transactions on Education*, vol. 53, no. 2, pp. 318–327, may 2010.

[21] M. Spitzer, "Lernen," *Gehirnforschung und die Schule des Lebens. Heidelberg: Spektrum*, 2002.

[22] C. Kröhn and B. Sabitzer, "Peer-learning and talents exchange in programming – experiences and challenges," *CSEDU, Prague, CZE*, 2020.