Informatics in Education, 2023, Vol. 22, No. 2, 183–206

DOI: 10.15388/infedu.2023.14

High-School Computer Science – Its Effect on the Choice of Higher Education

Michal ARMONI[1], Judith GAL-EZER[2]

[1]*Department of Science Teaching, Weizmann Institute of Science, Rehovot, Israel*
[2]*Mathematics and Computer Science Department, The Open University of Israel*
e-mail: michal.armoni@weizmann.ac.il, galezer@cs.openu.ac.il

Received: March 2022

Abstract. In a previous publication we examined the connections between high-school computer science (CS) and computing higher education. The results were promising – students who were exposed to computing in high school were more likely to take one of the computing disciplines. However, these correlations were not necessarily causal. Possibly those students who took CS courses, and especially high-level CS courses in high school, were already a priori inclined to pursue computing education. This uncertainty led us to pursue the current research. We aimed at finding those factors that induced students to choose CS at high school and later at higher-education institutes. We present quantitative findings obtained from analyzing freshmen computing students' responses to a designated questionnaire. The findings show that not only did high-school CS studies have a major impact on students' choice whether to study computing in higher education – it may have also improved their view of the discipline.

Keywords: post-secondary education, secondary education, computer science.

1. Introduction

Increasing numbers of countries have come to recognize the importance of pre-college computing education, and significant efforts to tap massive financial and human resources are currently being implemented to develop appropriate K-12 computing curricula (e.g., Corradini *et al.*, 2017; Lamprou *et al.*, 2017; The Royal Society, 2018; Syslo and Kwiatkowska; 2015; Falkner *et al.*, 2019; Anwar *et al.*, 2020; Dagienė *et al.*, 2021). The motivation behind these initiatives sets a few goals for including computer science (CS) in K-12 education. For the context of this work, we will focus on three of these goals.

One of the goals of teaching a scientific discipline at the K-12 level is to convey to the students a reliable image of the nature of the discipline (Armoni and Gal-Ezer, 2014a;). If students are exposed to a discipline in school, they have a basis for relying on later when considering a higher education track. When it comes to CS, such an exposure is even more important because of the inaccurate image common among the general public including students and their parents regarding the nature of the discipline of CS. There is often confusion between computer literacy, digital literacy, computer applications, or information technology and the scientific discipline of computer science (Carter, 2006; Mitchel *et al.*, 2009; Yardi and Bruckman, 2007; Wang *et al.*, 2016; Hewner, 2013). K-12 curriculum designers strive to introduce the students to the various facets of this scientific discipline, emphasizing that it constitutes much more than just programming or applications (Vogel *et al.*, 2017; Weilder-Lewis *et al.*, 2017; Webb *et al.*, 2017).

It is known that many students tend to hold negative attitudes towards CS, perceiving it, for example, as boring, asocial, fit for nerds, or not suitable for girls (Moorman and Johnson, 2003; Yardi and Bruckman, 2007; Leonard *et al.*, 2021; deWit *et al.*, 2021; Pantic *et al.*, 2018; Gurer *et al.* , 2019). Thus, a second goal is to positively affect students' attitudes towards CS, in particular, make them perceive CS as interesting and challenging. Curriculum developers also aim at making the students enjoy their CS high-school studies.

A third goal of CS K-12 studies that is reported in the literature (e.g., Armoni and Gal-Ezer, 2014a; Duncan and Bell, 2015; Guzdial *et al.*, 2014; Knobelsdorf *et al.*, 2014; Webb *et al.*, 2017) is motivating students to pursue computing studies at institutes of higher education.

These three goals are probably interconnected. Enjoying CS high-school studies and perceiving CS as interesting and challenging has a positive effect on the motivation to pursue computing at institutes of higher education. On the other hand, inaccurate or erroneous images of CS might negatively affect students' attitudes towards it and their decisions to study it in high school and at institutes of higher education.

We note that a central goal of CS K-12 education is to develop students' skills of algorithmic or computational thinking (e.g., Barr and Stephenson 2011; Grover and Pea 2013), which students can employ in other contexts, outside of CS. This goal is not the focus of our study; rather, our focus is the path from K-12 CS education to higher-education computing.

In Israel, which has a centralized school system, CS is only currently being introduced at the K-9 levels. However, a high-school CS curriculum has been implemented since the middle of the 1990s (Gal-Ezer *et al.*, 1995; Gal-Ezer and Harel 1999). Similar to other scientific disciplines taught in Israeli high schools, CS is an elective that had two versions, which differ regarding their depth and breadth: a basic 3-unit version (270 hours, three weekly hours throughout the three years of high school), designed for those students who have little interest in the discipline or want to acquire only some basic knowledge of what CS is; and an advanced 5-unit version, designed for those students who show more interest in the discipline (450 hours, five weekly hours throughout the three years of high school). In many schools, students who have taken advanced CS (5 units) can choose to continue to a 5-unit software engineering (SE) program, in which each student develops a software system of a specific kind (e.g., expert systems, operation systems, cyber security systems, and mobile phone systems).

The main notion that is emphasized throughout this curriculum is an algorithmic problem and its solution. Students are also exposed to many additional aspects of the discipline, beyond the implementation of algorithms in a programming language.

The implementation of the Israeli curriculum was accompanied by many studies that mostly assessed students' learning and understanding (e.g., Armoni and Gal-Ezer 2005, 2006; Ben-Ari and Ben-David Kolicant, 1999; Brewer, 1994; Gal-Ezer and Zeldes 2000; Lapidot *et al.*, 2000; Scherz and Habermann, 1995).

The curriculum and its implementation in its various aspects were considered successful in different senses: Relatively many students take it, with about the same number of students as those who take physics; Though the gender gap is prevalent, it is not as big as it is in other countries that implement a CS high-school curriculum; Students succeed in the matriculation examinations, which affect their acceptance into higher-education institutions (Armoni and Gal-Ezer 2014b; Hazzan *et al.*, 2008).

Research on the impact of K-12 (and specifically high-school) CS education on the rate of enrollment in higher-education computing programs is scarce. This motivated us to pursue our present research project. Since in Israel CS is currently taught at high schools, through a stable and successful program, we set out to examine the effects of CS high-school education, and specifically the connections between CS high-school education and computing at higher-education institutes. In a previous paper (Armoni and Gal-Ezer, 2014b) we were able to show promising results, including correlations between studying CS in high school and studying computing at institutes of higher education. Those students who were exposed to CS in high school, especially those who took high-level CS courses, were more likely to major in one of the computing disciplines (e.g., computer science, computer engineering, information systems, and software engineering) in their higher education studies. However, these correlations were not necessarily causal. Possibly those students who chose to take CS, and especially high-level CS in high school, were already inclined to pursue computing education, and this prompted them to choose both CS in high school and a computing discipline at institutes of higher education. Thus, we could not establish the direct impact of high-school CS on the decision to pursue computing at institutes of higher education.

One optional research direction would be to sample high-school CS graduates, to examine how many of them pursued computing at institutes of higher education, and to inquire into the reasons underlying why those high-school graduates who did pursue it further decided to do so. This route was not feasible, since we could not gain access to an identified pool of former high-school CS graduates. Therefore, we had to take another route. We decided to turn to CS freshmen, just starting their studies, and inquire about their high-school experience, and the factors underlying their decision to take computing at institutes of higher education. Retrospective methodology is used in science education research (Jones *et al.*, 2011), and retrospective questionnaires inquiring about school experience are considered a valid research tool (e.g., Carter, 2006).

This paper is organized as follows: In the next section we discuss relevant work. In Section 3 we describe the research plan, the research population, and the research tools. Then, in Section 4 we present our findings, and in Section 5 we conclude with a discussion of the findings and their consequences.

2. Related Work

As noted in the introduction, many countries have already developed or have started developing corresponding K-12 programs (see Section 1 for relevant references). A 2017 European report surveying the situation in over 50 European countries or autonomic regions found that in only three countries/regions was computing not offered at all in secondary schools, whereas in 33 countries/regions it was available (or even compulsory) for all students (Vahrenhold *et al.*, 2017). Some of these educational efforts have been or are being accompanied by research efforts with varying objectives. In this section we discuss the relevant studies. In line with the rationale of the current study, we mostly limited ourselves to those studies that deal with K-12 CS curricula, rather than extracurricular outreach initiatives.

In some places, direct access to K-12 students for research purposes is limited or even prohibited due to official regulations (e.g., Hubwieser, 2012); however, in other places there is active research on these aspects. Several of these studies aim at investigating students' learning and determining whether the knowledge and performance-related learning objectives were achieved (e.g., Armoni and Gal-Ezer, 2006; Bell *et al.*, 2014; Kert *et al.*, 2019). Generally, there are mixed results, some (e.g., Reppening *et al.*, 2015; Statter and Armoni, 2020) pointing to learning achieved, and some indicating partial success, with meaningful learning for some concepts but insufficient internalization for others (e.g., Armoni *et al.*, 2005; Meerbaum-Salant *et al.*, 2013).

For the purpose of this paper, we are mostly interested in studies that deal with three goals for integrating computing into K-12 curricula, those that we discussed in the introduction. Statter and Armoni (2020) were able to show that an introductory course for 7th graders positively affected *students' perception of CS*, with a significantly larger effect on girls (Statter and Armoni, 2017). Hildebrandt and Diethelm (2012) investigated the effect of a K-12 German curriculum on *students' interest in CS*. The only significant result was a small increase for girls. The third goal is concerned with *motivating students to pursue computing studies at higher education institutes*. There are two possible research directions for addressing this goal: the first examines students' intentions right after studying a K-12 computing course or program, and the second looks farther, at decisions rather than intentions, that is, it deals with the actual effect of K-12 computing education in the long run. Although it is important to study immediate effects, since they can indicate future intentions and decisions, the outcomes of such studies are inherently limited, since intentions may change. The potential contribution of the second research direction is obviously higher. Several researchers, though apparently not many, took the first direction. In the study by Hildebrandt and Diethelm (2012), mentioned above, they also investigated the aspect of choosing a professional career, but found no significant effect. Duncan and Bell (2015) conducted a preliminary study, aimed at assessing a pilot phase of the new K-12 computing program in New Zealand. Their findings indicated a high inclination towards pursuing computing, as an education or career path (though it was higher for boys than for girls), but since the motivation was not evaluated before the teaching process had begun, only after it, these results cannot shed light on the effect of the new computing program. Delyser (2014) examined the effects of a high-school software engineering curriculum. Future intentions were surveyed before and after the course regarding four aspects: learning software engineering or CS in school, learning software engineering or CS as an out-of-school activity, pursuing software engineering or CS in college, and choosing CS or software engineering as a career path. Students' intentions to participate in out-of-school activities or to pursue computing as a career path decreased following the course. Regarding students' intentions to pursue computing in school, there was a small decrease for boys and a slight increase for girls. Regarding students' intentions to pursue computing in college, there was a small increase for boys and a higher increase for girls. However, no information was given about statistical significance.

As for the second direction, since in most countries K-12 computing education is quite young, it is apparently too early to examine the long-term effects. Nevertheless, several studied pursued this line of research. Guzdial *et al.* (2012) examined how the GaComputes program influenced pursuing computing at higher education institutes in Georgia. GaComputes is a state-wide program for broadening participation in computing. It is not a curricular initiative that introduces a school curriculum in computing. Rather, it offers weekend workshops and summer camps in computing for children in late primary and secondary schools, as well as in-service professional development in computing for secondary school teachers. The authors surveyed over 1400 students in introductory computer science courses from 19 higher education institutes in Georgia, but more than half of these students were not computing majors. The collected data could indicate whether a student has studied in a school in which a teacher participated in GaComputes, but since the data provided no information regarding actual GaComputes-related previous experience, that is, whether the students were taught by or had any other connection with a teacher who participated in the program, the authors could not determine whether the program influenced the students' decision to take the introductory CS course, let alone a decision to major in computing . Similarly, although about 57% of the respondents took some computing course in high school (which was not necessarily GaComputes related), with 40% of the respondents reported taking a high-school course that included programming, the data could not allow for connecting high-school computing experience with the decision to take the introductory computer science course, let alone a decision to major in computing. The study illuminated additional factors, besides high-school computing experience, that had some influence on the choice to pursue computing at higher education institutes; this included the enjoyment of working with computers, future-career considerations, and perceived high abilities in math or science.

Hubwieser (2012) examined the pathway between the Bavarian K-12 CS program, which has been implemented since 2004, with the first cohort graduating in 2011, and computing at higher education institutes. In the Bavarian K-12 program, CS is taught from grade 6. It is a compulsory subject for all 6^{th}- and 7^{th}-grade students, a compulsory subject for all 9^{th}- and 10^{th}-grade students who take the science and technology track (one of four tracks, taken by about half of the students), and an elective subject in grades 11 and 12 of the science and technology track. Upon examining the background of 153 freshmen in an informatics faculty of one university, it was found that the percentage of those who have attended a CS course in a specific grade among these students is much higher than the corresponding percentage of a CS-course attendees in the entire population of students who were in that grade the same year. However, these findings do not indicate causal connections. For example, about 50% of the students who were in grades 9 and 10 in 2007 and 2008, respectively, took the corresponding CS course. Since this is a compulsory course in the science and technology track and this track is taken by about 50% of the students, this is not surprising. Among the 153 freshmen, this course was taken (in the same years) by about 78% of the students, but this may be because most of the freshmen in the informatics faculty from that university took the science and technology track. Regarding the 11th- and 12th-grades elective CS courses, the higher percentage can be explained by a possible early inclination towards CS, which has influenced both the choice to choose CS in grades 11 and 12 and in the university. Another limitation of this study is that only one university (out of 22 higher-education Bavarian institutes that offer a CS major) was investigated.

McGill *et al.* (2016) also examined the impact of pre-college activities on students' choice of major, as well on the students' perceptions of this choice. However, they did not distinguish between different kinds of activities, and hence the students could refer to curricular and extracurricular ones, without reporting on their nature.

The study described here deals with the long-term effects of K-12 computing education on students' perceptions of and attitudes towards the discipline and their decision to pursue computing in higher-education institutes. It adds to the work of Hubwieser (2012) and Guzdial *et al.* (2012) in several ways. First (unlike the study of Guzdial *et al.*), our studied population included only students who decided to pursue computing in higher education institutes (taking their first introductory CS course). Second, our data provided us with the students' individual experience in high-school CS education, as well as other relevant information, allowing us to examine possible connections between learning the high-school CS program in our country and students' decisions to pursue computing at higher education institutes.

3. The Research

3.1. *Research Objectives*

Motivated by the three goals of CS K-12 education detailed in Section 1, we had a few objectives: First, we wanted to investigate those factors that drove undergraduate computing students to choose a computing discipline (computer science, software engineering, computer engineering, and information systems, among others) at higher-education institutes. Second, we were also interested in revealing students' perceptions of the nature of CS as a discipline. Third, we aimed at revealing students' attitudes towards CS. For those students who had studied CS in high school, we were also interested in revealing the factors that affected their choice to take CS in high school, and their attitudes regarding the CS high-school curriculum. And finally, we were also looking for factors that affected high-school students choosing not to take CS courses in high school.

We wish to stress again that we had neither a specific research question nor a specific hypothesis. We wanted to collect as many data as we could and identify emerging phenomena. This impacted our research tool, as described below.

3.2. *Research Tool*

In order to achieve these objectives, we planned a corresponding questionnaire consisting of seven parts. Having no prior expectations or hypotheses, we aimed at taking a reliable "snapshot" that would depict things as they actually are. Therefore, each part contains a rich variety of items, intended to achieve a broad coverage of potential relevant factors or attitudes. To limit the questionnaire to be within a reasonable length, some of the questions were compound, examining multiple aspects. In addition, we repeated some of the questions in different contexts, that is, in different parts of the questionnaire, as we will elaborate in Section 4, which presents our findings.

The questionnaire included items regarding the reasons that prompted students to choose computing at higher-education institutes, as well as other items referring to reasons for taking or not taking high-school CS courses, students' attitudes towards high-school CS, and items regarding gender in the context of CS.

The seven parts are as follows:

1. Students' background: gender, CS high-school background – that is, whether they had taken high-school-level CS courses, the level of their CS high-school education (basic, advanced, or the SE track), elective modules, and final scores – and mathematics background (level and final score).
2. Factors that affected students' decisions to learn higher-education computing: 14 Likert-type items, with a scaling from 1 (definitely don't agree) to 5 (very much agree). For example, "I chose to study a computing program because I was interested in this field since I was very young" or "I chose to study a computing program because I studied CS in high school and found it interesting".
3. (For students who had studied CS in high school) Factors that affected the students' decision to learn CS in high school: 13 Likert-type items, scaled as above. For example, "I chose to take CS in high school because programming attracts me".
4. (For students who had not studied CS in high school) Factors that affected students' decision not to learn CS in high school: seven Likert-type items, scaled as above. For example, "I chose not to take CS in high school since I did not know what CS is".
5. (For students who had studied CS in high school) Attitudes towards their high-school CS experience: 13 Likert-type items, scaled as above. For example, "I enjoyed studying CS in high school" or "High-school CS made me think about taking an undergraduate CS program".
6. Interest in the elective modules taken as part of their CS high-school studies: Three Likert-type items, one for each elective module, with scaling from 1 (not at all interesting) to 5 (very interested).
7. Attitudes towards gender and CS: seven Likert-type items, scaled as above. For example, "Females succeed in CS studies just as well as males do".

3.3. *Population*

We planned to administer the questionnaires to undergraduate computing students. We aimed at minimizing the effects of the undergraduate studies on students' responses, since we were interested in their initial attitudes and in factors that had potentially affected their previous decision to enroll in an undergraduate computing program. Therefore, we decided to administer the questionnaire to students taking the first introductory course (CS1) and to do so as early as possible, no later than the second week of the semester, preferably during the first week.

After obtaining the approval of the IRB (Institutional Review Board), we distributed the questionnaires to students in seven introductory CS classes, enrolled in four universities and one college (which was recently acknowledged as a university). In these five institutions about 800 students were enrolled in seven CS1 classes. The questionnaires were administered and immediately filled out during class. The total estimation of 800 students was based on the estimated non-tight upper bounds given to us by these five institutions. In all these institutions there was no formal requirement to attend classes, so there were probably fewer students attending class while we distributed the questionnaire, but we do not have the exact number. Nevertheless, we received responses (non-empty questionnaires) from 427 students.

Of the 427 students, three did not report their gender. Of the other 424 students, 71.70% were males and 28.30% were females (see Fig. 1). This ratio is similar to the overall ratio (70.85% males and 29.15% females) among all the students who pursued higher-education computing in the years 1995–2011, as obtained from the Israeli Bureau of Statistics.

The students' average age was about 23. Most students (92.84%) were between 18 and 27 years old, but there were students as old as 49 and as young as 17 (eight students did not report their age).

3.4. *Students' High-School CS Background*

We asked the students whether they had studied CS in high school. Four hundred and twenty-four students (all but three) answered this question: 267 (62.97%) of these students had studied CS in high school, and 157 students (37.03%) had not studied CS in high school (Fig. 2).

Four hundred and twenty-one students (all but six, 302 males and 119 females) reported both their gender and their CS high-school experience: 199 males and 66 females had studied CS in high school. Thus, 75.19% of the students that took CS in high school were males and 24.91% of the students that took it in high school were females (Fig. 3a). From another point of view, 65.89% of the 302 male students took CS in high school (Fig. 3b) and 55.46% of the 119 female students took CS in high school (Fig. 3c).

Two hundred and sixty-six students who had studied CS in high school (all but five) reported the level of their CS high-school course. Most of them, 248 students,

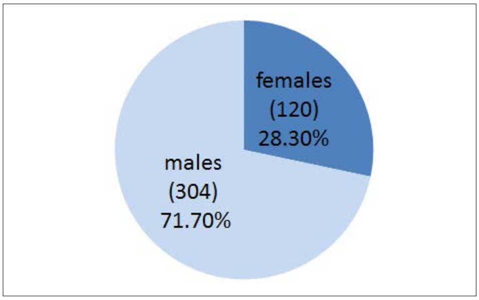

Fig. 1. Entire population by gender.

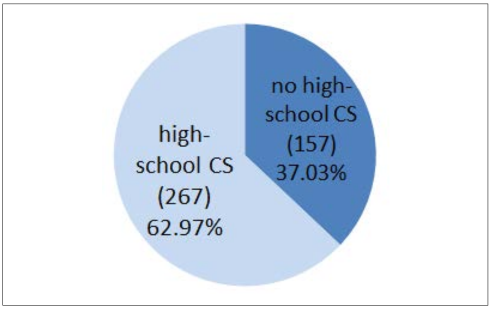

Fig. 2. Entire population by high-school CS experience.

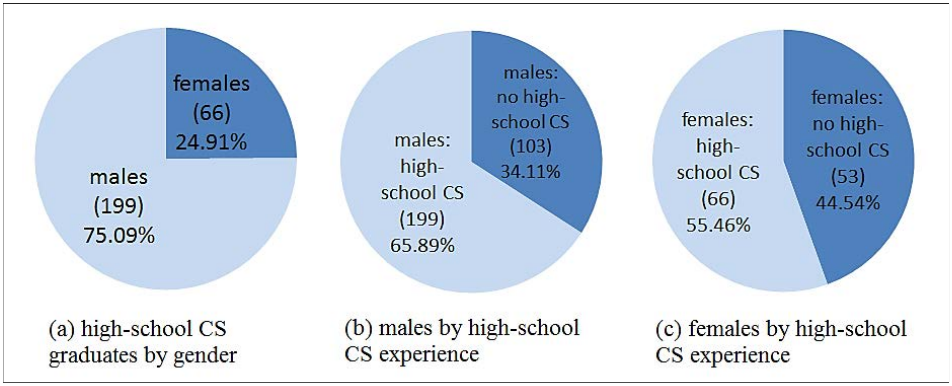

Fig. 3. Entire population by high-school CS and gender.
(a) High-school CS graduates by gender.
(b) Male students by high-school CS experience.
(c) Female students by high-school CS experience.

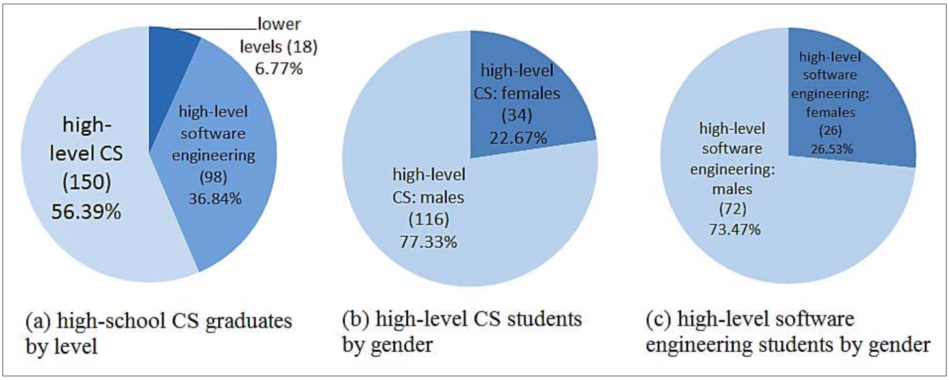

Fig. 4. High-school CS graduates by level and gender.
(a) High-school CS graduates by level.
(b) High-level CS students by gender.
(c) High-level software engineering students by gender.

chose to specialize in CS in high school. One hundred and fifty (56.39%, Fig. 4a) took a high-level (in terms of breadth as well as depth) CS course (about 77% of them males and about 23% females, Fig. 4b). Another 98 students (36.84%, Fig. 4a) also added a high-level course in software engineering (about 73% males and 27% females, Fig. 4c).

4. Findings

This section presents the findings, as obtained by analyzing students' answers on the questionnaire. It is organized as follows: Section 4.1 deals with the reasons for choosing to study computing at higher-education institutes (the second part of the questionnaire). In Sections 4.2 and 4.3 we present the findings concerning the reasons for choosing or not choosing to study computer science in high school (the third and fourth parts of the questionnaire, respectively). Section 4.4 deals with students' attitudes towards their CS high-school experience (the fifth part of the questionnaire). Section 4.5 does not correspond to a specific part of the questionnaire. Rather, it looks at students' perceptions of the nature of CS, as reflected by their answers to several questions taken from different parts. Finally, Section 4.6 deals with students' attitudes towards gender in the context of CS (the seventh part of the questionnaire).

4.1. *"Why Did I Choose Computing as an Undergraduate Major?"*

This part of the questionnaire included 14 items. We report here only on 13 of the 14 items. We realized that one of the items was not phrased properly, and no meaningful information could be extracted from the students' responses in this case; therefore, we

Table 1
Reasons for choosing computing at institutes of higher education

Ranking	Item number	Item	N	Average score	S.D.
1	12	I find this field interesting.	392	4.49	0.72
2	8	I thought I would succeed in a future computing job.	397	4.34	0.80
3	11	I thought I could succeed.	396	4.31	0.78
4	9	Programming attracts me.	395	4.22	0.88
5	6	A computing career is financially rewarding.	396	4.15	0.87
6	13	It is a prestigious field.	391	4.05	0.93
7	3	I was interested in this field since I was very young.	398	3.72	1.19
8	4	CS contains a central mathematical component.	397	3.44	1.16
9	7	My parents recommended it.	385	2.74	1.23
10	2	It is a scientific, yet not an engineering subject, so I thought it would be appropriate for girls.	294	1.58	0.98
11	1	Friends of mine chose to study it.	381	1.55	0.84

Table 2
CS high-school experience as a factor in choosing to study computing at institutes of higher education

Item number	Item	N	Average score	S.D.
5	I studied CS in high school and found it interesting.	252	4.27	0.95
10	I studied CS in high school and succeeded.	250	4.22	1.06

decided to omit it. Two of the items were relevant only to those students who had studied CS in high schools, and they will be reported later (Table 2). Table 1 presents, therefore, the average scores for the remaining 11 items.

Eight of the 11 items (above the thick line) were acknowledged, on average, as having a positive effect on the decision to choose computing at institutes of higher education (with average scores higher than 3), and three (below the thick line) were not considered as factors that affected the choice of computing at higher-education institutions (the average score was lower than 3). Among the three factors that were not considered as positive reasons, two were related to an external influence (Item 7, “my parents recommended it”, and Item 1 “friends of mine chose to study it”). Another was a gender-related item.

The highest average score was for Item 12, “I find this field interesting”. From a wider perspective, the top seven items concerned interest, success (after graduation or during the undergraduate studies), and attributes of a future career (prestigious or financially rewarding). Items 12, 9, and 3 refer to interest. Items 8 and 11 refer to success. Items 6 and 13 refer to profit – financial profit, or a more affective profit of prestige. Item 4, indicating recognition of the mathematical facet of CS, was ranked last of all ten items with positive average scores. We will revisit this issue of CS and mathematics and will elaborate on it in Section 4.5.

Table 2 presents the results for the two items that were relevant only for students who had studied CS in high school. It clearly shows that high-school CS studies had a major impact on choosing to study computing at institutes of higher education.

Table 3 distinguishes those who had studied CS in high school and those who had not. Comparing the results for these two groups, interest in the field is the strongest reason for both, and the overall ranking is similar, except regarding attraction to programming. We can see that for eight of the 11 items, the highest average score was among those students who had studied CS in high school; the items for which the differences were statistically significant are shaded in gray. This may indicate a higher motivation for studying computing at higher-education institutes, among those students who had studied CS in high school, since the same factors affected them more strongly. This higher motivation could have developed during their high-school CS studies, or perhaps they were a priory more inclined to study CS.

Our findings offer support for both scenarios. The gap between the two groups was especially large regarding interest in the field since a young age. This indicates the early inclination of the students who had studied CS in high school. However, *their high-school experience might have strengthened this prior inclination*, since among their reasons for choosing computing at institutes of higher education, the items regarding high-school experience (Table 2) had a higher impact than the impact of the interest in the field since a young age. In Section 4.2 we elucidate this issue further, comparing the factors affecting students' choice in high school and at higher-education institutes, respectively.

Table 3

Reasons for choosing computing at institutes of higher education – CS high-school graduates and non-CS high-school graduates, non-high-school-related items (*– p < 0.0001; **– p < 0.05)

# item	Item	CS high-school graduates				Non-CS high-school graduates			
		Ranking	N	Average score	S.D.	Ranking	N	Average score	S.D.
12**	I find this field interesting.	1	248	4.56	0.72	1	141	4.37	0.71
9*	Programming attracts me.	2	251	4.39	0.82	6	141	3.91	0.91
11**	I thought I could succeed.	3	251	4.37	0.83	3	142	4.19	0.68
8	I thought I would succeed in a future computing job.	4	251	4.36	0.86	2	143	4.30	0.70
6**	A computing career is financially rewarding.	5	251	4.22	0.87	4	142	4.03	0.86
13	It is a prestigious field.	6	249	4.08	0.95	5	139	3.98	0.88
3*	I was interested in this field since I was very young.	7	251	3.99	1.09	8	144	3.24	1.20
4	CS contains a central mathematical component.	8	250	3.41	1.19	7	144	3.50	1.12
7	My parents recommended it.	9	246	2.78	1.24	9	136	2.63	1.21
2	It is a scientific, yet not an engineering subject, so I thought it would be appropriate for girls.	10	184	1.53	0.97	11	108	1.66	1.02
1**	Friends of mine chose it.	11	238	1.46	0.75	10	140	1.71	0.96

4.2. *"Why Did I Choose CS in High School?"*

The third part of the questionnaire was directed to those students who had studied CS in high school. It included 13 Likert-type items scaled from 1 (definitely don't agree) to 5 (very much agree). In Table 4 we present the average scores of the items of this part.

The first three items relate to students' interest in the discipline, whereas the next three items look farther into a professional career and its potential advantages (desirable, prestigious, and rewarding – financially and also in the sense of appreciation). This is somewhat similar to the students' considerations regarding undergraduate studies, as depicted in Table 1, only here items of both kinds are not interwoven. The items with the highest impact refer to interest, followed by those items relating to a future professional career. Bearing in mind that this is retrospective data, this indicates that these students, at the age of 16, valued interest considerations more than they valued future career considerations. Yet, it is clear that even at the age of 16, they did not ignore direct and indirect future professional considerations, when making decisions regarding the choice of CS. It was interesting to note that factors that relate to external influence (parents, friends, and advisors) and factors that relate to CS in the context of their entire high-school program (such as the relative difficulty of CS compared to other subjects) are in the lower part of the table, not perceived to be influential (on average).

A few items appear in both the second (Table 3) and third (Table 4) parts of the questionnaire. It is interesting to examine these pairs, comparing the effect of the same factor in the context of high-school and undergraduate education, correspondingly. We present here the comparison for a few of these pairs, and others, which relate to the nature of the discipline, will be addressed in Section 4.5.

Table 4

Reasons for choosing CS in high school

Ranking	# item	Item	N	Average score	S.D.
1	1	I thought it was an interesting subject.	255	4.43	0.87
2	3	Programming attracts me.	255	4.02	1.06
3	2	I was interested in computers since I was very young.	252	3.97	1.20
4	7	CS is a desirable profession and one can make a lot of money in it.	252	3.80	1.16
5	6	CS is an appreciated and prestigious profession.	254	3.75	1.23
6	5	I knew I would like to work in this field in the future.	253	3.70	1.22
7	4	CS is similar to Math.	254	3.22	1.25
8	8	It was a required subject in my track.	230	2.70	1.54
9	13	My parents recommended it.	244	2.40	1.33
10	10	I thought it would be a relatively easy subject compared to other scientific subjects.	248	2.13	1.35
11	12	Friends of mine chose it.	241	1.80	1.13
12	11	There wasn't any other high-level subject that I wanted to take.	242	1.77	1.16
13	9	My advisor recommended it.	226	1.69	1.08

- **Early inclination:** Interest in the field since a young age was evaluated similarly when considered as a reason for choosing CS in high school and as a reason for choosing computing at higher-education institutes. This indicates that the early inclination to study CS was stable for this population. Yet, as noted in Section 4.1 and depicted in Table 2, their high-school CS experience strengthened this prior inclination.
- **Interest in the discipline:** Interest in the field was evaluated as a factor with high impact in both cases, for choosing CS in high school and for choosing computing at higher-education institutes. However, its average score was higher for the case of higher education with a statistically significant difference ($p < 0.02$). *We can therefore conclude that high-school CS had a positive impact, since the interest in the discipline of those students who had studied CS in high school grew significantly*.
- **Future career:** Three pairs of items relate to a future career. Table 5 compares the average scores for these items when considered as reasons for choosing CS in high school and at higher-education institutes, correspondingly (for all three pairs the differences were significant, $p < 0.0001$).

This is consistent with our findings presented above, according to which at the age of 16, the impact of future career considerations on choosing CS was substantial, yet smaller than the impact of interest considerations.

- **External influence:** External influence, whether of parents or of friends, was not considered as an impacting factor for choosing CS, in high school and at higher-education institutes. However, the influence of friends was higher regarding high school than regarding higher-education institutes. ($p < 0.0001$). Indeed, in high school the societal effect when choosing a subject to study is probably stronger than for more mature persons. The influence of parents was higher regarding higher education than regarding high school. This can be explained by differences in attitudes towards parents in adolescence and as adults.

Table 5

Factors relating to future careers, regarding choice of CS in high school and at institutes of higher education

High-school choice of CS					Undergraduate choice of computing				
#	Item	N	Average score	S.D.	#	Item	N	Average score	S.D.
5	I knew I would like to work in this field in the future.	253	3.70	1.22	8	I thought I would succeed in a future computing job.	251	4.36	0.86
6	CS is an appreciated and prestigious profession.	254	3.75	1.23	13	It is a prestigious field.	249	4.08	0.95
7	CS is a desirable profession and one can make a lot of money in CS.	252	3.80	1.16	6	A computing career is financially rewarding.	251	4.22	0.87

Taken together, these findings indicate that the factors that positively affected students' choice of CS in high school became even more pronounced as factors that motivated their choice of computing at institutes of higher education. *This suggests that high-school CS studies had a positive impact regarding the choice of computing as a learning and career path.*

4.3. *"Why Did I Not Choose CS in High School?"*

A little more than a third of our research population – 157 students – reported that they did not choose CS in high school. Note that in spite of their previous choice, they did choose to pursue one of the computing programs at institutes of higher education. The fourth part of the questionnaire (see Table 6) was directed to these students (though not all of them filled it out). It included seven Likert-type items, scaled from 1 (definitely don't agree) to 5 (very much agree), each of which is a possible reason for not choosing CS in high school.

The only item for which the average score was above 3 is Item 4, "Other school subjects interested me more than CS did". The next three highest items were Items 1, 2, and 5, with average scores of 2.84, 2.43, and 2.34, respectively. Indeed, a non-negligible number of students gave these items a positive score (4 or 5).

These findings relate to the global concerns of our community, as well as of education policy makers and curriculum designers. Students who were not methodologically exposed to CS at an earlier stage in school as for other scientific disciplines, probably would not know what CS is, and what its various facets consist of. Therefore, they might think it is boring, and less interesting than other subjects. In addition, if CS does not count for credit for graduation from high school, as is the case in some countries, or if it counts less than other school subjects, students will prefer other scientific subjects that are more rewarding, especially when it comes to being admitted to a computing program in a prestigious higher-education institute.

These issues should be addressed when designing a CS school curriculum, and indeed, they are a concern that our community shares.

Table 6

Reasons for not choosing CS in high school

Ranking	# item	Item	N	Average score	S.D.
1	4	Other school subjects interested me more than CS did.	128	3.78	1.33
2	1	I did not know what CS is.	126	2.84	1.37
3	2	I thought it was a boring subject.	125	2.43	1.23
4	5	I found it more important to take another scientific subject that would help me get admitted more easily to a good university.	117	2.34	1.23
5	6	I do not like programming.	117	2.00	1.08
6	7	It has too much math.	118	1.70	0.83
7	3	I did not think it was a good subject for girls.	91	1.67	0.94

4.4. *“How Do I Feel about High-School CS?”*

The fifth part of the questionnaire focused on students’ attitudes towards their high-school CS studies. It was directed to students who had studied CS in high school. It included 13 Likert-type items with scaling from 1 (definitely don’t agree) to 5 (very much agree). In Table 7 we present the average scores of the items of this part. Note that some of the items have a compound structure. This was done to keep the length of the questionnaire within reasonable limits. The compound structure might have induced ambiguity. However, we could triangulate these items with the first item, to clarify their interpretations.

These findings indicate that at least two of the three goals (mentioned in Section 1) of a high-school CS program were achieved: The students enjoyed studying CS in high school (Item 1), they thought it was interesting (Items 2 and 3) and challenging (Item 13), and it positively affected their choice to study computing at institutes of higher education (Item 5).

As noted in the introduction, another goal of high-school CS curriculum designers is to expose high-school students to the various facets of the discipline. A few items in this fifth part of the questionnaire relate to this goal. The next section is devoted to elucidating the nature of CS as viewed by the students.

In addition, our findings indicate that the role of teachers is significant and has an impact on students’ experience with high-school CS (Item 4). Although this seems a straightforward statement, it reinforces the importance of establishing appropriate teacher preparation programs for pre-service CS teachers, as well as investing in pro-

Table 7

Attitudes towards high school CS (CS high-school graduates)

	# item	Item	N	Average score	S.D.
1	1	I enjoyed studying CS in high school.	234	4.40	0.89
2	8	I enjoyed high-school CS studies because programming interests me.	232	4.09	1.04
3	5	High-school CS made me think about taking an undergraduate CS program.	233	3.91	1.11
4	4	I enjoyed high-school CS because the teacher was very good.	234	3.47	1.27
5	13	I enjoyed high-school CS since it was challenging.	218	3.42	1.04
6	11	I enjoyed high-school CS since it was easy for me.	231	3.34	1.24
7	7	High-school CS studies do not represent the field properly.	232	2.81	1.10
8	6	High-school CS studies dealt with topics that were different from those I had expected.	230	2.80	1.14
9	3	High-school CS was not as interesting as I had expected.	234	2.16	1.15
10	2	High-school CS was boring.	232	2.14	1.16
11	9	I did not enjoy [studying CS in high school] since it was too similar to mathematics.	230	1.80	0.89
12	12	I did not enjoy high-school CS since I had to devote a lot of time to it.	224	1.72	0.83
13	10	I did not enjoy high-school CS since I already knew programming.	218	1.67	0.87

fessional development programs for in-service teachers. The Computer Science Teacher Association (CSTA) and others make enormous efforts to set standards for CS teaching (CSTA, 2016); also many European states have slowly but surely come to understand the importance of teachers' pre-service and in-service training (Hazzan *et al.*, 2010; Informatics Europe and ACM Europe, 2013; The Royal Society, 2012; Sentance and Csizmadia, 2017).

4.5. *Perception of the Nature of CS*

As emphasized in Section 1, an important goal of school CS curriculum designers is to introduce the students to the various facets of this scientific discipline. That is, emphasizing that it constitutes much more than just programming or applications, and that it also has theoretical, even math-flavored facets. Our results can shed light on the issue of high-school CS graduates' perception of CS.

One of the high-scored items concerning the reasons for choosing computing at institutes of higher education was attraction to programming (Item 9, Table 1). The average score of this item was 4.22, and it was graded 4th among the 13 items. This indicates that when considering computing as a higher-education track, many students based their decision on their interest in programming and on the assumption that programming is at least a significant and major component of CS. CS high-school graduates scored and ranked this item higher, compared to those who had not studied CS in high-school (Table 3). When considered as a reason for choosing to study CS in high school, attraction to programming was also scored and ranked high, but not as high as for choosing computing at higher-education institutes.

The above might suggest that after completing CS in high school, the attraction of programming became an even stronger factor for studying CS. This conjecture might be strengthened by the high scores that those students who had studied CS in high school gave Item 8 of the attitudes part, "I enjoyed high-school CS studies because programming interests me" (Table 7).

Was this because after studying CS in high school these students had a stronger view of CS as programming? Students' attitudes towards the connection between CS and math illuminate this issue from a different angle. As a reason for choosing CS in high school, the similarity of CS and mathematics (Item 4, Table 4) had a positive (though not high) average score. That is, on average, this item was acknowledged as an impacting factor. When these students, who had studied CS in high school, were asked to consider the mathematical component of CS as a reason for choosing computing at institutes of higher education (Item 4, Table 3), the average score of this item was higher ($p < 0.008$). This finding might indicate that after completing CS in high school, these students had a stronger view of CS as having a mathematical component, or that the mathematical component of CS had a stronger impact on their decision to study CS.

Judging by their attitudes towards high-school CS studies (Table 7), we can assume that acknowledging the mathematical component of CS probably did not have a negative effect on their high-school CS experience, since the similarity of CS and mathematics did not affect their enjoyment of studying CS in high school (Item 9).

This result suggests that CS high-school studies had a positive impact on creating a more reliable image of CS. Moreover, since these students viewed CS as more than just programming, in its standard narrow interpretation, their own interpretation of the term "programming" might be a wider one.

In line with the objective of the Israeli high-school CS program, and the goals of curriculum designers in general, of introducing CS to students as a scientific discipline that is much more than just programming, and which also has theoretical, even math-flavored facets, the curriculum includes a variety of topics and points of view. In particular, the extended high-school CS course offers an opportunity to learn such topics. As noted in Section 3, more than 93% of the students who had studied CS in high school took the extended course (and some of them augmented it by taking a high-level software engineering course). Part of the extended course had a few alternatives, of which the teachers had to choose one to teach. More than 60% of the students who had studied CS in high school (N = 160) reported on which alternative they had studied. Fig. 5 depicts the distribution of these students over the five alternatives. The alternative "computational models" is different from the other alternatives. This is a 90-hour theoretical unit, which does not involve any kind of programming, and introduces the students to the mathematically flavored area of automata and formal languages (including some topics in computability theory).

There was no significant difference in students' attitudes towards their CS high-school experience, regarding the above-mentioned alternatives. In particular, this is true for the items that directly relate to the nature of CS (see Items 6, 7, and 9 in Table 7). However, there was one exception, the item "I enjoyed high-school CS studies because programming interests me" (see Item 8 in Table 7). The average scores of the students who had studied "computational models" (4.05) is lower ($p<0.05$) than those that had studied "advanced object-oriented programming" (4.38). Since both groups of students

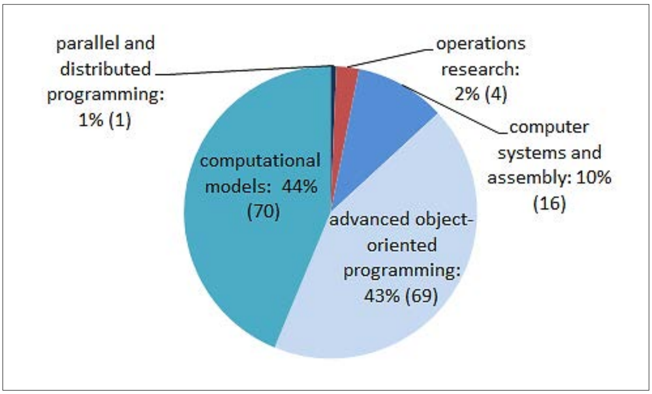

Fig. 5: High-school high-level-CS graduates by alternatives

enjoyed high-school CS (with average scores of 4.60 and 4.70, respectively, and no statistically significant difference), there are at least three possible interpretations:

1. The students who had studied "computational models" tended less to identify CS with programming.
2. The students who had studied "computational models" were interested in programming, but still it did not affect their enjoyment just as much.
3. The students who had studied "computational models" were less interested in programming, but it did not affect their enjoyment. In all cases apparently, they were aware of CS including components other than programming.

Taken together, our findings indicate the impact of high-school CS on students' views of the nature of CS, and on their understanding that CS has more in it than just programing in its narrowest sense.

4.6. *Gender and CS*

The seventh part of the questionnaire focused on attitudes towards gender and CS. It includes seven Likert-type items, with scaling from 1 (definitely don't agree) to 5 (very much agree). For example: "Females succeed in CS studies just as well as males do".

As depicted in Table 8, students ranked very low those items that argue that there is a gap between males and females when it comes to their abilities to learn and practice CS, whereas items that argue against such a gap were ranked very high. Item 6 ("Most of my peer students are males"), which was also ranked high, did not refer to perceived abilities; rather it examined the state of affairs regarding the number of females who study CS.

Bearing in mind that about 2/3 of the respondents are males, we find these findings very encouraging, in the sense that they may insinuate that an evolution is taking place, at least regarding the attitudes of younger people towards women and computing disciplines, as opposed to known misconceptions and prejudices concerning this issue, which are still widespread. Nevertheless, these findings confirm what is known from the literature (and are supported by the make-up of our population) that currently, fe-

Table 8

Attitudes towards gender in the context of CS

Ranking	# item	Item	N	Average score	S.D.
1	1	CS is a profession in which females can succeed.	342	4.60	0.73
2	4	Females succeed in CS studies just as well as males do.	336	4.15	0.96
3	6	Most of my peer students are males.	327	4.03	1.00
4	5	Females successfully fit in the Hi Tech industry.	327	4.01	0.92
5	3	CS is not a profession for females because it demands long working hours.	326	1.68	0.94
6	2	CS is not a profession for females because it is technological.	319	1.46	0.89
7	7	CS is not for females because it involves a lot of math.	322	1.42	0.74

males represent a minority when it comes to choosing to study CS at institutes of higher education (e.g., Medel and Pournaghshband, 2017; Zweben and Bizot, 2021).

The important issue of gender can also be illuminated by reexamining the previous parts of the questionnaire through the prism of gender. This yields an extensive and interesting analysis, which will be reported in a separate publication.

5. Discussion

As mentioned in the introduction, in a previous study, motivated by the question whether it is worthwhile to encourage students to take CS courses in high school, we showed encouraging results; students who took CS in high school, especially those who took high-level CS courses, were more likely to pursue computing at institutes of higher education. This was even more pronounced regarding female students. Following this research, we wanted to learn more regarding the impact of high-school CS on students' choice of computing at higher-education institutes. In addition, we wanted to reveal the attitudes of computing students who had studied CS in high school towards high-school CS. This motivated us to conduct the current research on which we reported here. Indeed, as we presented in detail in the previous sections, we identified several positive effects of high-school CS on students' attitudes towards and perceptions of CS, and their choice of a computing discipline at higher-education institutes. We will discuss our main insights in what follows. Owing to the limitations of the very few studies that aimed at addressing these questions or some of them, our findings clearly contribute to the body of research-based literature on the outcomes of K-12 CS curricula.

5.1. *High-school CS as a Pipeline to Higher Education*

From our research population – students who were in the first class of the introductory CS course in four universities and one college – 60% had studied CS in high school. This can be easily concluded from students' responses to the first part of the questionnaire, but even more interesting and encouraging conclusions can be drawn from the other parts of the questionnaire that we described in detail in the previous sections.

Our findings indicate that high-school CS studies had a positive impact on choosing to study computing at institutes of higher education. In general, CS high-school graduates pointed out interest and success in their CS high-school studies as significant reasons for pursuing computing at higher-education institutes, and reported that high-school CS made them think about enrolling in an undergraduate CS program.

5.2. *Attitudes towards High-school CS*

Clearly students that studied CS in high school enjoyed it. Not surprisingly, the teachers also played a significant role by positively affecting students' enjoyment of their CS high-school studies. This is an important finding. When the Israeli CS curriculum was designed, the committee made a very clear point: “Teachers certified to teach the subject [i.e., CS] must have an adequate formal CS education. An undergraduate degree in computer science is a mandatory requirement, as is formal teacher training” (Gal-Ezer *et al.*, 1995, p.76). This was not obvious at the time and is still not obvious today in some countries. Indeed, teachers constitute the cornerstone for implementing any educational program. As mentioned in Section 4.4, CSTA and others make many efforts to set up standards for CS teaching, and European countries slowly but surely have come to understand the importance of teachers’ pre-service and in-service training. In Israel the importance is clear. The establishment of a special center for CS teachers followed the introduction of the high-school program. Nowadays the center supports all teachers, whether in high, junior high, or elementary schools. The center, called Machashva1 (the Hebrew word for “thought”), maintains a website that includes, among other things, resources of different kinds and forums for teachers; it organizes workshops and professional development courses, as well as invites computer scientists and educators to keynote presentations at annual conferences. Its main function is to encourage the teachers to contribute to each other’s knowledge based on their experiences regarding methodologies and pedagogical issues, especially when the curricula are updated and new material is introduced.

Students’ choice not to take CS in high school may also be connected to their attitudes towards CS at the time they made this choice. Since eventually these students chose to major in computing in their undergraduate studies, it is interesting to examine how they explained their reluctance to take CS in high school. The most influential factor was CS being less interesting than other subjects. A similar (yet absolute rather than relational) factor, “CS is boring”, scored positively by an appreciable number of students. That is, these students formed an opinion about the interest level of CS without ever having been exposed to it as a school subject. In addition, an appreciable number of students felt that they did not know what CS is. This shows that students who had to decide regarding whether to study CS in high school, and had not been exposed to CS earlier, like they were exposed to other sciences (Physics and Biology) could have no idea what it is about, and thus could not make a rational choice regarding the sciences they wanted to learn in high school. Therefore, it is recommended to introduce CS at least in middle school if not earlier (CSTA, 2006, 2016; Informatics Europe and ACM Europe, 2013; The Royal Society, 2012).

A few students found it more important to take another scientific subject that would help them get admitted more easily to a good university. The issue of accrediting CS similarly to the other sciences is an issue that has been discussed worldwide. In Israel educators and university policy makers debate whether CS is on the same par as the other sciences and whether students should receive the same bonus with it when enrolling in a university. In this age, when the importance of CS is increasingly recognized worldwide, policy makers should pay attention to the credit students receive for studying CS in high school when they later apply to high-education institutions.

[1] `http://cse.proj.ac.il/` in Hebrew

5.3. *Gender in the Context of CS*

We were happy to see that students ranked very low those items that argue that there is a gap between the abilities of males and females when it comes to CS, whereas items that argue against such a gap were ranked high. These findings, and especially when one bears in mind that about 2/3 of the respondents were males, may insinuate that an evolutionary process is taking place. As mentioned previously, an analysis of the entire questionnaire through the prism of gender yielded many interesting and meaningful results on which we intend to report in another publication.

5.4. *The Nature of CS as a Discipline*

We did not have a separate part in the questionnaire that discussed the nature of CS as a discipline, but we could form conjectures from the responses of the students, which shed some light on the issue of their perception of the nature of the discipline.

Internationally many initiatives have tried to put computing on the agenda for policy makers, stakeholders, and parents. However, computing (or informatics, as it is called in Europe) is not always introduced properly, and often only one facet of the discipline is introduced, namely, programming or coding. This may stem from the misconception that CS is programming, or from the misconception that programming or rather coding is the only facet of CS that school students can understand, or the desire of educators to attract more students to CS. This can lead to an incorrect image of the discipline and can result in a negative effect on whether students decide to major in CS in high school or at higher-education institutes.

When designing the Israeli high-school curriculum, the committee emphasized that “the program should concentrate on the key concepts and foundations of the field” and that “conceptual and experimental issues should be interwoven throughout the program”, which was termed the “zipper principle”. However, as mentioned in Section 4.5, we noted that although attraction to programming received a high average score as a factor in choosing CS in high school, its average score was even a bit higher as a factor in choosing a computing discipline later. Moreover, for students who had studied CS in high school, the average score was even higher. We also noted that students who had studied CS in high school attributed their enjoyment of high-school CS to their interest in programming. All this might insinuate that the curriculum was not implemented as the designers had intended. Apparently, after studying CS in high school, students got the incorrect impression that CS equals programming.

On the other hand, if we examine the mathematical component of CS as a reason for choosing CS in high school, we see that it was acknowledged as an impacting factor. When students who had studied CS in high school were asked to evaluate the mathematical component of CS as a factor for choosing computing as an undergraduate major, the average score was even higher. That is, these students consider the mathematical component as central to CS, and the impact of this acknowledgment increased after they studied CS in high school. This insinuates a positive impact of high-school CS on students’ understanding of what CS really is.

5.5. *Summary*

To summarize, our findings indicate that taking CS program in high school increased students' motivation to study one of the computing disciplines when pursuing higher education. They also indicate that high-school CS also had a positive impact on the students' views of the nature of CS, and on their understanding that CS is more than just programming.

Consequently, we can positively respond to the question whether high-school CS has a positive impact on students' decision making; therefore, we will continue our efforts to place CS as a scientific discipline on the same par as the other disciplines already accredited for students' admission to universities. Our results can also assist in designing future curricula for CS in schools or in updating the existing ones.

Acknowledgement

We thank:

The Research Authority of the Open University of Israel for supporting this research;
Yetti Varon for the statistical analysis;
Yael Bachar for the conversion of the raw data into standard quantitative format.

References

Anwar, T., Jimenez, A., Bin Najeeb, A., Upadhyaya, B., McGill, M.M. (2020). Exploring the Enacted Computing Curriculum in K-12 schools in south Asia: Bangladesh, Nepal, Pakistan, and Sri Lanka. In: *Proceedings of the 2020 International Computing Education Research Conference* (ICER'20). ACM press, New York, 79–90.

Armoni, M., Gal-Ezer, J. (2005). Teaching reductive thinking. *Mathematics and Computer Education,* 39(2), 131–142.

Armoni, M., Gal-Ezer, J. (2006). Introducing non-determinism. *Journal of Computers in Mathematics and Science Teaching*, 25(4), 325–359.

Armoni, M., Gal-Ezer, J. (2014a). Early computing education – Why? What? When? Who? *ACM Inroads Magazine*, 5(4), 54–59.

Armoni, M., Gal-Ezer, J. (2014b). High school computing education paves the way for higher education and assists in closing the equity gap – the Israeli case. *Computer Science Education,* 24(3), 101–122.

Armoni, M., Gal-Ezer, J., Tirosh, D. (2005). Solving problems reductively. *Journal of Educational Computing Research,* 32(2), 113–129.

Barr, V., Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of computer science education community? *ACM Inroads,* 2(1), 48–54.

Bell, T., Andreae, P., Robins, A. (2014). A case study of the introduction of computer science in NZ schools. *ACM Transactions on Computing Education,* 14(2), 10: 1–31.

Ben-Ari, M., Ben-David Kolikant, Y. (1999). Thinking parallel: the process of learning concurrency. In *Proceedings of the 4th ACM Annual Conference on Innovation and Technology in Computer Science Education* (ITiCSE1999). ACM press, New York, 13–16.

Brewer, W. (1994). Autobiographical memory and survey research. In: Schwarz, N., Sudman, D. (Eds.), *Autobiographical Memory and the Validity of Retrospective Reports*. Springer-Verlag, New York, 11–20.

Carter, L. (2006). Why students with an apparent aptitude for computer science don't choose to major in computer science. In: *Proceedings of the 38th Technical Symposium on Computer Science Education* (SIGCSE2006). ACM Press, New York.

Corradini, I., Lodi, M., Naedelli, E. (2017). Computational thinking in Italian schools: quantitative data and teachers' sentiment analysis after two years of "programma il futuro". In: *Proceedings of the 22nd ACM Conference on Innovation and Technology in Computer Science Education* (ITiCSE2017). ACM Press, New York, 224–229.

CSTA (2006). *Running on Empty: The Failure to Teach K-12 Computer Science in the Digital Age*. ACM Press, New York. http://runningonempty.acm.org/fullreport2.pdf

CSTA (2016). *Interim CSTA K-12 Computer Science Standards (Revised).* ACM Press, New York.

Dagienė, V., Hromkovič, J., Lacher, R. (2021). Designing informatics curriculum for K-12 education: From concepts to implementations. *Informatics In Education*, 20(3), 333–360.

DeLyser, L. (2014). Software engineering students in the city. In: *Proceedings of the 9th Workshop in Primary and Secondary Computing Education* (WiPSCE2014). ACM Press, New york, 37–42.

Duncan, C., Bell, T. (2015). A pilot computer science and programming course for primary school students. In: *Proceedings of the 10th Workshop in Primary and Secondary Computing Education* (WiPSCE2014). ACM Press, New York, 39–48.

Falkner. K., Sentance, S., Vivian, R., Barksdale, S., Busuttil, L., Cole, E., Liebe, C., Maiorana, F., McGill, M., Quille, K. (2019). An international comparison of K-12 computer science education intended and enacted curricula. In: *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (Koli Calling'19). ACM Press, New York, 4:1–10.

Gal-Ezer, J., Beeri, C., Harel, D., Yehudai, A. (1995). A high-school program in computer science. *Computer*, 28(10), 73–80.

Gal-Ezer, J., Harel, D. (1999). Curriculum and course syllabi for a high school CS curriculum. *Computer Science Education*, 9(2), 114–147.

Gal-Ezer, J., Zeldes, A. (2000). Teaching software designing skills. *Computer Science Education*, 10(1), 25–38.

Grover, S., Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, 42(1), 38–43.

Gurer, M.D., Cetin, I., Top, E. (2019). Factors affecting students' attitudes toward computer programming. *Informatics in Education*, 18(2), 281–296.

Guzdial, M., Ericson, B.J., McKlin, T., Engelman, S., (2012). A statewide survey on computing education pathways and influences: Factors in broadening participation in computing. In: *Proceedings of the 9th Annual International Conference on International Computing Education Research* (ICER2012). ACM Press, New York, 143–150.

Guzdial. M., Ericson, B., McKlin, T., Engelman, S. (2014). Georgia computes! An intervention in a US state, with formal and informal education in a policy context. *ACM Transactions on Computing Education*, 14(2), 13:1–29.

Hazzan, O., Gal-Ezer, J., Blum, L. (2008). A model for high school computer science education: The four key Elements that make it! In: *Proceedings of the 40th Technical Symposium on Computer Science Education* (SIGCSE2008). ACM Press, New York, 281–285.

Hazzan, O., Gal-Ezer, J., Ragonis, N. (2010). How to establish a computer science teacher preparation program at your university, the ECSTPP workshop. *ACM Inroads Magazine*, 1(1), 35–39.

Hewner, M. (2013). Undergraduate conceptions of the field of computer science. In: *Proceedings of the 10th Annual International Conference on International Computing Education Research* (ICER2013). ACM Press, New York, 107–114.

Hildebrandt, C., Diethelm, I. (2012). The school experiment InTech – How to influence interest, self-concept of ability in Informatics and vocational orientation. In: *Proceedings of the 7th Workshop in Primary and Secondary Computing Education* (WiPSCE2012). ACM Press, New York, 30–39.

Hubwieser, P. (2012). Computer science education in secondary schools – the introduction of a new compulsory subject. *ACM Transactions on Computing Education*, 12(4), 16:1–41.

Informatics Europe, ACM Europe (2013). Informatics education: Europe cannot afford to miss the boat. http://www.informatics-Europe.org/images/documents/informatics-education-acm-ie.pdf

Jones, G., Taylor, A., H. Forrester, J.H. (2011). Developing a scientist: A retrospective look. *International Journal of Science Education*, 33(12), 1653–1673.

Kert, S.B., Kalelioğlu, F., Gülbahar, Y. (2019). A Holistic approach for computer science education in secondary schools. *Informatics in Education*, 18(1), 131–150.

Knobelsdorf. M, Magenheim, J., Brinda, T., Engbring, D., Humbert, L., Pasternak, A.,

Schroeder, U., Thomas, M., Vahrenhold, J. (2014). Computer science education in North-Rhine Westphalia, Germany – a case study. *ACM Transactions on Computing Education*, 15(2).

Lamprou, A., Repenning, A., Escherle, N. (2017). The Solothurn project – bringing computer science education to primary schools in Switzerland. In: *Proceedings of the 22nd ACM Conference on Innovation and Technology in Computer Science Education* (ITiCSE2017). ACM Press, New York, 218–223.

Lapidot, T., Levy, D., Paz, T. (2000). Teaching functional programming for high school students. In: *Proceedings of the International Conference on Mathematics/Science Education and Technology (M/SET2000)*. AACE, Charlottesville, 245–249.

Leonard, H., Quinlan, O., Sentance, S. (2021). Female pupils' attitudes to computing in early adolescence. In: *Proceedings of the United Kingdom and Ireland Computing Education Research conference* (UKICER'21). ACM Press, New York, 5:1–6.

McGill, M.M., Decker, A., Settle, A. (2016). Undergraduate students' perceptions of the impact of pre-college computing activities on choices of major. *ACM Transactions on Computing Education*, 16(4), 15:1–33.

Medel, P., Pournaghshband, V. (2017). Eliminating gender bias in computer science education materials. In: *Proceedings of 49th Technical Symposium on Computer Science Education* (SIGCSE2017). ACM Press, New York, 411–416.

Meerbaum-Salant, O., Armoni, M., Ben-Ari, M. (2013). Learning computer science concepts with Scratch. *Computer Science Education*, 23(3), 239–264.

Mitchell, A., Purchase, H.C., Hamer, J. (2009). Computing science: what do pupils think? In: *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education* (ITiCSE2009). ACM Press, New York, 353–353.

Moorman, P., Johnson, E. (2003). Still a stranger here: Attitudes among secondary school students towards computer science. In: *Proceedings of the 8th Annual ACM* SIGCSE *Conference on Innovation and Technology in Computer Science Education (ITiCSE2003)*. ACM Press, New York, 193–197.

Pantic, K., Clarke-Midura, J., Poole, F., Roller, J., Allan, V. (2018) Drawing a computer scientist: stereotypical representations or lack of awareness? *Computer Science Education*, 28(3), 232–254.

Repenning, A., Webb, D.C., Koh, K.H., Nickerson, H., Miller, S., B., Brand C., …, Repenning, N. (2015). Scalable Game Design: a strategy to bring systemic computer science education to schools through game design and simulation creation. *ACM Transactions on Computing Education*, 15(2), 11:1–31.

The Royal Society. (2012). *Shut Down or Restart? The Way Forward for Computing in UK Schools*. The Royal Society, London.

The Royal Society. (2018). *After the Reboot: Computing Education in UK Schools*. The Royal Society, London.

Scherz, Z., Haberman, B. (1995). Logic programming based curriculum for high school students: the use of abstract data types. In: *Proceedings of 26th Technical Symposium on Computer Science Education* (SIGCSE1995). ACM Press, New York, 331–335.

Sentance, S., Csizmadia, A. (2017). Professional recognition matters: certification for in-service computer science teachers. In: *Proceedings of 49th Technical Symposium on Computer Science Education* (SIGCSE2017). ACM Press, New York, 537–542.

Statter, D., Armoni, M. (2017). Teaching abstraction in computer in 7th grade – the perspective of gender. In: *Proceedings of the 12th Workshop in Primary and Secondary Computing Education* (WiPCSE2017). ACM Press, New York. 5–14.

Syslo, M., Kwiatkowska, A., B. (2015). Introducing a new computer science curriculum for all school levels in Poland. In: *Proceedings of the 8th International Conference on Informatics in Schools (ISSEP2015)*. Springer, 141–154.

Vahrenhold, J., Nardelli, E., Pereira, C., Berry, G., Caspersen, M.E., Gal-Ezer, J., Kölling, M., McGettrick, A., Westermeier, M. (2017). *Informatics Education in Europe: Are We All In The Same Boat?* ACM Europe/ Informatics Europe.

Vogel, S., Santo, R., Ching, D. (2017). Visions of computer science education: Unpacking arguments for and projected impacts of CS4All initiatives. In: *Proceedings of 49th Technical Symposium on Computer Science Education* (SIGCSE2017). ACM Press, New York, 609–614.

Wang, J., Hong, H., Ravitz, J., Hejazi Moghadam, S. (2016). Landscape of K-12 computer science education in the U.S.: Perceptions, access, and barriers. In: *Proceedings of 48th Technical Symposium on Computer Science Education* (SIGCSE2016). ACM Press, New York, 645–650.

Weidler-Lewis, J., Dubow, W., Kaminsky, A. (2017). Defining a discipline or shaping a community: constraints on broadening participation in computing. In: *Proceedings of 49th Technical Symposium on Computer Science Education* (SIGCSE2017). ACM Press, New York, 627–632.

Widdicks, K., Ashcroft, A., Winter, E., Blair, L. (2021). Women's sense of belonging in computer science education: The need for a collective response. In: *Proceedings of the United Kingdom and Ireland Computing Education Research conference* (UKICER'21). ACM Press, New York, 4:1–7.

de Wit, S., Hermans, F., Aivaloglou, E. (2021). Children's implicit and explicit stereotypes on the gender, social skills, and interests of a computer scientist. In: *Proceedings of the 2021 International Computing Education Research Conference* (ICER'21). ACM press, New York, 239–251.

Yardi, S., Bruckman, A. (2007). What is computing?: Bridging the gap between teenagers; perceptions and graduate students; experiences. In: *Proceedings of the 3rd International Workshop on Computing Education Research (ICER2007).* ACM Press, New York, 39–50.

Zweben, S., Bizot, B. (2021). 2020 Taulbee survey: Bachelor's and doctoral degree production – Growth continues but new students enrollment shows decline. *Computing Research News*, 33(5), 2–68.

M. Armoni is an associate professor at the Department of Science Teaching, Weizmann Institute of Science. She received her PhD from the School of Education in Tel-Aviv University, and her B.A. and M.Sc. in computer science (CS) from the Technion, Israel Institute of Technology. She has been engaged in CS education for more than 20 years as an educator, a curricular developer, and a researcher. She has co-authored several CS textbooks for high schools and middle schools. Her research focuses on teaching and learning processes of CS, through the prism of CS fundamental ideas.

J. Gal-Ezer is a professor of computer science at the Open University of Israel (OUI), the department of Mathematics and Computer Science. She served as head of the department, head of the Development division, and Vice President for Academic Affairs at the OUI. After several years of research on *wave propagation* and *seismology* (as part of her M.Sc. and Ph.D. studies), her research activity turned to two directions: One – the teaching of *computer-integrated mathematics*; the other – *computer science education*, which became her main area of interest. She was on the Ministry of Education professional committee which put together the CS curriculum for Israeli high-schools. She later chaired this committee. She is the recipient of ACM SIGCSE 2007 "Special Contribution to computer science education, the IEEE 2015 Computer Society Taylor L. Booth Education Award "For outstanding research and its practical application in the field of computer science education", ad the 2017 ACM Karl V. Karlstrom Outstanding Educator Award. Since 2022 she is Fellow of the ACM.