HAL
open science

Does Studying CS Automatically Foster a Growth Mindset?

Michael Lodi

To cite this version:

Michael Lodi. Does Studying CS Automatically Foster a Growth Mindset?. ITiCSE 2019 - 24th ACM Conference on Innovation and Technology in Computer Science Education, Jul 2019, Aberdeen, United Kingdom. pp.147-153, 10.1145/3304221.3319750 . hal-02379130

HAL Id: hal-02379130

https://inria.hal.science/hal-02379130v1

Submitted on 25 Nov 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Does Studying CS Automatically Foster a Growth Mindset?

Michael Lodi*
University of Bologna & INRIA
Department of Computer Science and Engineering (DISI)
Bologna, Italy
michael.lodi@unibo.it

ABSTRACT

Many arguments are used to advocate for the introduction of Computer Science (CS) / Computational Thinking / "coding" in K-12 education. Growth mindset theory (GM) is also becoming very popular among educators and researchers. Some claims stating that studying CS can foster a GM have emerged. However, educational research shows that transfer of competences is hard. Very little research has been conducted on the relationship between GM and CS learning, with conflicting results. We measured some indicators (e.g., mindset, computer science mindset) at the beginning and the end of a high school year in five different classes: three CS oriented, one Chemistry oriented, and one Transportation&Logistics oriented. In one of the CS oriented classes, we did a very brief GM intervention. At the end of the school year, none of the classes showed a statistically significant change in their mindset. Interestingly, non-CS oriented classes showed a significant decrease in their computer science growth mindset. In the intervention class, students suggested, to stimulate a GM, the need for activities that are more creative, engaging, and related to the real world and their interests.

CCS CONCEPTS

• **Social and professional topics → Computing education**; **K-12 education**; *Computational thinking*.

KEYWORDS

computer science mindset, growth mindset, high school, transfer

ACM Reference Format:
Michael Lodi. 2019. Does Studying CS Automatically Foster a Growth Mindset?. In *Innovation and Technology in Computer Science Education (ITiCSE '19), July 15–17, 2019, Aberdeen, Scotland, UK*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3304221.3319750

1 INTRODUCTION

Recently the introduction of Computer Science (CS) / Computational Thinking (CT) / "coding" in K-12 education has gained massive attention from researchers, educators, and media. Many arguments are used to advocate this introduction, e.g., understanding our digital world, being active and creative digital citizens, learning skills for digital jobs, and learning to "think better." However, some of them seem not to be based on sound educational research (see the recent [18] for a good review).

At the same time, mindset theory [10] (as it is widely known today, more formally called "implicit theories about intelligence" [7]) is becoming very popular among educators and researchers. People hold personal ideas (*implicit theories*) about their intellectual abilities. These ideas lay on a continuous spectrum, but two main groups can be identified. Some people have a *fixed mindset* (an *entity theory of intelligence*): they think their intelligence is a fixed aspect, and they cannot do much to change it. By contrast, other people have a *growth mindset* (an *incremental theory of intelligence*): they think intelligence is malleable and can be developed with study and effort. Some studies show that students' mindset can influence their achievement [4]. Moreover, Dweck showed that a growth mindset could be fostered with brief specific interventions [4] or specific teaching strategies (e.g., giving growth mindset feedback [9]). Dweck acknowledges that one can have different mindsets with respect to different disciplines [10]: in this paper, we will refer to the *mindset with respect to CS* as *Computer Science mindset*.

Some claims stating that studying CS/CT/programming/"coding" can foster a GM have emerged: partially because of the popularity in educational contexts, partly because some characteristics of Computer Science - and in particular of programming (e.g., collaboration, iterative development, continually finding and correcting errors) - fit well in the mindset theory. The thesis that learning computing can foster a growth mindset is exemplified by some quotes from online educational blogs. "*Not only can the process of learning to code be used to help develop a growth mindset, there are aspects of coding that help reinforce this mindset*" [34]. "*[...] its contribution to a growth mindset, is one of the reasons why debugging is such an important part of computing education*" [3]. "*But did you know that by learning to code, you can simultaneously grow your growth mindset in the process?*" [36][1]. "*Over three years of helping kids teach themselves computer programming, I have decided that youth coding programs are the best way to instill a growth mindset in another human being*" [32]. "*Posing challenging tasks such as basic coding even at the elementary level pays great dividends and fosters a Growth Mindset among even the youngest students*" [13].

Other CS Ed. researchers seem to acknowledge the problem. For example, Lewis states, talking about growth mindset and grit, that "*programming has been speculated to be uniquely qualified to help normalize failure and thus encourage productive learning strategies*" [18]. As Lewis herself recognizes, research in education tells

*Part of this research was conducted while the author was visiting Prof. Tim Bell and the Computer Science Education Research Group at the Department of Computer Science and Software Engineering, University of Canterbury, Christchurch, New Zealand.

[1] Another blog post on Mindsetworks also seems to acknowledge the opposite direction of the implication: Having a Growth Mindset helps learning to code [35].

ITiCSE '19, July 15–17, 2019, Aberdeen, Scotland, UK
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Innovation and Technology in Computer Science Education (ITiCSE '19), July 15–17, 2019, Aberdeen, Scotland, UK*, https://doi.org/10.1145/3304221.3319750.

us that transfer is difficult and unlikely to happen, especially between knowledge domains far from one another [2]. Research has shown that it is possible to teach transfer, if the activities are explicitly designed to do so: for example, in the 1980s, in the context of LOGO programming, some researchers were able to explicitly teach debugging and to transfer it to "buggy" situations outside programming (e.g. in wrong road directions or furniture arrangement in a room) [17]. Note however that these contexts are much more similar to the original programming one than perseverance or growth mindset are. Research about transfer in Computer Science education (also regarding transversal competencies/skills) is recognized to be contradictory and outdated [27].

To better understand the relationship between studying or not studying Computer Science and mindset (and other related constructs, especially CS mindset), we decided to conduct a study in a high school with different tracks. One track is Computer Science oriented, while others are oriented to different technical skills (e.g., Chemistry, Logistics and Transportation). With our experiment, we tried to address the following research questions:

RQ1 Does studying or not studying Computer Science in high school automatically influence students' mindset, Computer Science mindset, and other related constructs?

RQ2 Can a small explicit intervention alter students' mindset?

RQ3 After being taught about mindset theory, what are students' recommendations to foster it?

2 COMPUTER SCIENCE AND GROWTH MINDSET

2.1 Growth Mindset

Students[2] with a growth mindset show **learning-oriented goals** (not afraid to ask and make errors, in order to learn) and **mastery-oriented responses** (greater effort and new strategies) to challenges and setbacks, while students with a fixed mindset show **performance goals** ("appear intelligent", so avoiding difficult tasks) and **helpless responses** to challenges (e.g. giving up or blaming the teacher for their failure).

In many studies, Dweck and colleagues showed that students' mindset could predict their achievement, in particular in Math and Science, and their ability to cope with challenges [4, 9]. Moreover, female students with a growth mindset showed less susceptibility to the adverse effects of stereotypes about women and math [12].

Some interventions [9] can positively stimulate a growth mindset: explicitly teaching students about mindsets, brain plasticity and the idea that intelligence can be trained with effort; portraying challenges, effort and mistakes as highly valued; praising process and effort (e.g., "You worked so hard and did an excellent job!") rather than person or talent ("Good! You must be very smart!"), and giving constructive feedback rather than praising the person or being judgmental.

2.2 Growth Mindset in Computer Science

In [22] it is argued that a growth mindset can be particularly important in CS education. For example, in programming, students are continually facing errors and challenges; female enrollment in CS is low, possibly due to fixed views about who can and who cannot succeed in CS; collaboration is fundamental in CS and is fostered by growth mindset students whose goal is to learn rather than to appear smart, and so on. Factors like repeated exposure to errors and stereotypes are recognized to potentially foster a fixed mindset, as has been suggested for math [5]. In [6] is stated that Carol Dweck herself describes CS as a discipline that requires a growth mindset.

Very few studies assessed students' mindset before and after a CS course (at a college level). In [11] the authors observed changes in CS1 students (CS-majors, other STEM-Majors, but also Arts and Business Majors) across the semester, finding a significant increase in a fixed mindset and a significant decrease in a growth mindset. Moreover, in [28] it was found that beliefs about intelligence and programming aptitude form two distinct constructs in undergraduate software engineering students. The mindset for programming aptitude had greater utility in predicting software development practice, and a follow-up survey showed that it became more fixed after a programming module. In [15], third-year bachelor students' mindset was assessed, finding no correlations between their mindset and their grades.

Even fewer studies have tried to alter students' mindset, and these have found conflicting results. The authors in [30] tried a small intervention on CS Majors and Minors, finding a shift toward fixed mindset despite the intervention. In contrast, the authors in [6] performed three structured interventions into an introductory programming course, gaining significant improvement in growth mindset level of students and also a positive correlation in their test scores.

We could not find any studies about Computer Science and growth mindset at the high school level.

3 THE STUDY

3.1 Context

In Italy, high school (formally "upper secondary school") lasts five years (usually students start when they are 14 y.o. and finish when they are 19 y.o.). Students can choose between **lyceums**, which give a theoretical basis in classical, scientific or artistic areas and naturally lead to university studies; **technical institutes**, which give both theoretical basis and a high qualified specialization in a specific area (divided in: *economic institutes*, preparing for economics, management and business-oriented computer programming, and *technological institutes*, preparing for areas such as mechanics, electronics, computer science, chemistry), leading to a job or to university studies; or **professional institutes**, preparing students with practical skills to enter the job market immediately.

After choosing the kind of school, students select a specific track offered by the school (e.g., "IT and telecommunications," "fashion"), whose curriculum is decided at national level. Almost every curriculum includes Italian, English, History, Math, Natural Sciences, Sports, and the subjects of the chosen track.

3.2 Participants

We conducted the experiment in four classes (third year, 16-year-old students) of a large technological institute (in brackets we give a short code we will use to identify the classes): 2 classes of "IT and

[2]Part of this section is an expansion of those in [19, 20].

Table 1: Descriptive statistics of the sample

Class	N	Males	Females	Prev. exp
INF1	16	16	0	7
INF2	14	11	3	1
INF3	7	7	0	7
TRAN	6	6	0	3
CHEM	23	16	7	12
ALL	*66*	*56*	*10*	*30*

telecommunications" (INF1, INF2); 1 class mixed "IT and telecommunications" (INF3) and "transport and logistics" (TRAN) (students follow intersecting subjects together and split when following specific subjects); and 1 class of "chemistry, materials and biotechnology" (CHEM).

During the first two years (usually very similar, to allow pupils to change track), all students of a technological institute should have followed some introductory computing lessons (no more than 2-3 hours a week for one or two years), but the content varies a lot: from the use of office suites to visual programming in Scratch to web development. We asked students if they had already programmed, to check for potential effects.

During the third year, students in "IT and telecommunications" follow three courses related to Computer Science and programming (Network systems, IT project management, Computer science) for a total of 13 hours a week. Students in "transport and logistics" and "chemistry, materials and biotechnology" do not follow any CS-related course.

We asked for an informed consent from students' parents, and only those who gave us back the approved informed consent participated in the study. As the study involved a pre and a post questionnaire, students were anonymously identified with a secret code (the name of a color) they had to remember. We report, in Table 1, number, gender and previous programming experience of only students that completed both the pre and the post-questionnaire. Most of the students are males, and the females are almost all from the CHEM class. This is not surprising since technological tracks, especially those related to mechanics, electronics and computer science, are usually chosen mainly by males in Italy.

3.3 Methods

We decided to measure some constructs (some belonging to general mindset theory, e.g., performance vs. mastery orientation - see subsection 2.1 - and some specifically related to CS mindset) at the beginning and at the end of the third year (school year 2017/2018). Moreover, we decided to implement a small mindset intervention in the INF1 class, halfway through the year, to see if we could alter students' mindset.

3.3.1 Questionnaire. Research on mindsets is usually conducted by asking students to rate how much they agree (on a Likert scale, e.g., 1-6) on some statements like "*You have a certain amount of intelligence, and you can't really do much to change it.*" and then calculating a score representing students' mindset. In the online appendix of this paper [21], the full questionnaire is reported, with details on how the scores are calculated.

For traditional mindset constructs, robust and validated scales are available. We used the wording in [8], an Italian translation of Dweck's book [7], to measure:

- **Mindset**, obtaining a value from 1 (fixed mindset) to 6 (growth mindset);
- **Confidence in one's intelligence**, obtaining a value from 1 (low confidence) to 12 (high confidence);
- **Goal choice**, obtaining a value from 1 (performance orientation) to 6 (mastery orientation).

To measure belief constructs related to CS, we adapted and translated in Italian some scales from [33], a Ph.D. thesis studying specifically Mathematical Mindsets, changing the word "Math" to "Computer Science" in the questionnaire proposed by the author. In particular, we measured:

- **Computer Science Mindset**, obtaining a value from 1 (fixed CS mindset) to 6 (growth CS mindset);
- **Beliefs about the nature of CS**, obtaining a value from 1 (CS as a fixed discipline, where you have to learn facts, rules, give quickly the *only* right solution to problems) to 6 (CS as a creative discipline, where you can learn from your errors and have many solutions to problems);
- **Identification with CS**, obtaining a value from 1 ("not a CS person", bad at CS) to 5 ("a CS person", good at CS);
- **Mastery orientation in CS**, obtaining a value from 1 (performance orientation in CS) to 5 (mastery orientation in CS).

The school was unable to authorize access to students' grades. However, we asked CS-oriented students if their grades in CS-related disciplines decreased, increased or remained the same.

The questionnaire was administered to all classes in two days, one at the beginning and one at the end of the school year, in school laboratories, through Google Forms, by the author (who has no relationships with the school, and presented himself as an entirely external subject, to avoid the tendency to socially desirable answers given by students to their teachers).

3.3.2 Intervention. In class INF1, halfway through the year, we performed a mindset intervention, attempting to alter students' beliefs by explicitly teaching them about growth mindset (analogously to what Dweck and colleagues did in many experiments: see for example [4]). The intervention (a 2 hours lesson conducted by the author, who is a computer scientist with a background in education) was inspired by material prepared by Khan Academy and PERTS [16]. The lesson included, in order: a discussion on what students think intelligence is, and if it could be changed; a discussion about a situation where they learned to do something they were not able to do yet; a presentation of the Mindset theory through a table translating a diagram from Nigel Holmes [14]; a short video [29] on neuroplasticity (with Italian subtitles); and a description by the researcher of "a situation where he overcame a struggle in learning and learned to solve a problem" [16].

In the last part of the lesson, three open questions were administered through a Google Form. Students were asked to think of a

Table 2: Internal consistency (Cronbach's alphas)

Measure		INF1	INF2	INF3	CHEM	TRAN	ALL
GM	pre	0.86	0.90	0.66	0.70	0.89	0.83
	post	0.94	0.97	0.93	0.82	0.26	0.89
CONF	pre	0.62	0.87	-0.34	0.81	0.49	0.67
	post	0.71	0.79	0.17	0.60	0.07	0.65
GOAL	pre	0.62	0.50	-0.21	0.46	-0.19	0.46
	post	0.35	0.29	-0.39	0.37	0.59	0.30
CSM	pre	0.60	0.80	0.57	0.71	0.52	0.72
	post	0.53	0.79	0.82	0.77	0.80	0.75
NAT	pre	-0.43	-0.21	0.13	-0.56	0.05	-0.08
	post	0.30	0.27	0.73	-0.52	0.51	-0.01
IDCS	pre	0.88	0.84	0.70	0.87	0.60	0.86
	post	0.91	0.77	0.76	0.83	0.74	0.85
MAST	pre	-0.43	0.69	0.81	0.35	0.49	0.40
	post	0.21	0.60	0.75	0.66	0.31	0.57

time when they overcame a struggle to learn something and then to:

- OQ1 Advise future students on how to overcome an obstacle when learning something new (translation of question in [16])
- OQ2 Suggest what a teacher could do to stimulate a growth mindset in students (proposed by us)
- OQ3 Suggest some concrete strategies to a future CS student to succeed and not be discouraged when facing difficulties in learning to program (inspired by [30])

The full wording of the questions is available in the online appendix of this paper [21].

4 RESULTS

4.1 Quantitative analysis

Responses were analyzed through the statistical programming language R [25], with the packages *Hmisc*, *pastecs*, *lsr*, *psych*, *dplyr*.

Internal consistency was measured through Cronbach's alpha [23] for questions about mindset (GM), confidence in one's own intelligence (CONF), goal choice (GOAL), Computer Science mindset (CSM), beliefs about nature of CS (NAT), identification with CS (IDCS), mastery orientation in CS (MAST) for pre- and post-questionnaire responses, for each class and for the whole sample of students. Values are reported in Table 2.

For many measures the internal reliability is very low: it could be due to the relatively small sample size, or to the translation/adaptation to CS of some questions, or other factors. An in-depth analysis of the reasons is out of the scope of this work. We decided to explore further with the analysis of GM, CSM and IDCS, that have acceptable (> 0.70) Cronbach's alphas [23] for both pre and post test for combined samples.

4.1.1 Difference between beginning and end of the year. To check if studying or not studying CS had an impact, we performed statistical tests on the difference between pre and post scores for each class and each of the three measures. Moreover, we also performed the test for all the samples combined.

Paired samples t-test works if the differences between pre and post scores are normally distributed. We checked that with the Shapiro-Wilk normality test and, since often it gives false positive results for small samples, also checked normality graphically through QQ-plots. Whenever we were unsure about the normality of data, we performed a non-parametric version of the test, in particular, the Wilcoxon signed rank test with continuity correction, that does not require a normality assumption. When we found significant results, we also measured the effect size through Cohen's d for paired samples. Results are reported in Table 3.

Mindset decreased in INF1, INF3, CHEM and increased in INF2 and TRAN, but in no case was the change statistically significant. Regarding the combined samples, mindset decreased for both CS and non-CS students, but not significantly.

Identification with CS decreased in INF classes, and increased in CHEM and TRAN, but not significantly in any case. This reflects on the fact that it decreases for CS students combined, and increases for non-CS students (not significantly, but with $p = 0.10$).

Computer Science mindset decreased, but not significantly, in INF1 and INF2, remained stable in INF3, but decreased significantly ($p = 0.02$) in CHEM (with a medium effect size $d = 0.52$) and decreased ($p = 0.09$, with a large effect size $d = 0.87$) in TRAN. In the combined samples, CSM decreased, but not significantly, in CS students, and significantly in non CS students ($p < 0.01$, with a medium/large effect size $d = 0.59$).

4.1.2 Difference between different subgroups. To check if there were significant differences between some subgroups, we performed many independent samples tests. In particular, we checked for differences in GM and CSM (at the beginning, at the end, and their difference) between males and females, between students with and without previous programming experience, and between students of CS and non-CS classes.

To perform an independent samples t-test, both subgroups must be normally distributed and have the same variance. All normal subgroups were checked with the F-test to compare the two variances and were found to have the same variance. Groups that were not normally distributed were analyzed with Wilcoxon rank sum test with continuity correction, which does not require normality assumption. Results are shown in Table 4.

We found no statistically significant differences in mindset between any of the subgroups. By contrast, we see significant differences in the difference between final and initial CS growth mindset. In particular, it decreased in females more than in males ($p = 0.16$), decreased much more in novices than in those with previous experience ($p = 0.06$) and, confirming previous observations, decreased statistically significantly more in non-CS students than in CS students ($p = 0.03$).

4.1.3 Correlation with grades. To see if mindset influences CS grades, we calculated Pearson's r between GM/CSM (pre, post, diff) and the reported variation in grades, but didn't find significant correlations, as shown in Table 5.

4.2 Qualitative analysis

We analyzed the responses to open questions from the students in INF1. Although the questions asked them to write a letter or

Table 3: Paired samples tests

Sample	Measure	n	normal	pre		post		t-test			Cohen's d	Wilcoxon-test	
				mean	sd	mean	sd	t	df	p		V	p
INF1	GM	16	yes	4.02	1.28	3.33	1.39	1.56	15	0.14			
	CSM	16	not sure	4.50	0.84	4.20	0.97					38.00	0.31
	IDCS	16	yes	3.28	0.96	3.07	0.94	1.02	15	0.32			
INF2	GM	14	yes	4.05	1.54	4.17	1.69	-0.36	13	0.72			
	CSM	14	not sure	4.68	1.02	4.38	1.12					58.50	0.38
	IDCS	14	not sure	3.71	0.65	3.68	0.83					47.00	0.94
INF3	GM	7	yes	4.29	1.15	4.05	1.46	0.35	6	0.74			
	CSM	7	not sure	3.50	0.80	3.50	1.14					8.50	0.89
	IDCS	7	not sure	3.83	0.46	3.75	0.55					10.00	0.59
CHEM	GM	23	yes	4.22	1.07	3.99	1.22	0.98	22	0.34			
	CSM	23	yes	4.21	1.11	3.71	1.12	2.51	22	0.02*	0.52		
	IDCS	23	yes	2.79	0.83	2.91	0.88	-0.83	22	0.42			
TRAN	GM	6	no	2.89	1.34	3.22	0.78					2.00	0.79
	CSM	6	yes	5.04	0.62	4.33	0.90	2.14	5	0.09	0.87		
	IDCS	6	not sure	2.59	0.45	3.20	0.44					1.00	0.06
ALL CS	GM	37	yes	4.08	1.33	3.78	1.53	1.14	36	0.26			
	CSM	37	no	4.38	0.98	4.14	1.08					259.50	0.20
	IDCS	37	yes	3.54	0.79	3.43	0.88	1.18	36	0.25			
ALL NON-CS	GM	29	yes	3.94	1.23	3.83	1.17	0.55	28	0.59			
	CSM	29	yes	4.38	1.07	3.84	1.10	3.18	28	0.00*	0.59		
	IDCS	29	yes	2.75	0.76	2.97	0.81	-1.71	28	0.10			

to give suggestions in at least five sentences, all responses were brief and schematic. This, however, made the coding procedure straightforward: we identified categories of advice and could easily fit answers in those categories.

We analyzed the responses (see sec. 3.3.2) of the 16 students of INF1 that also completed both the pre and the post questionnaires.

Categories and the number of suggestions fitting in that category are reported here. Note that many answers contained more than one piece of advice, so fit in more than one category.

OQ1 Categories: put effort (4); do not discourage (2); practice a lot (2); be convinced before choosing CS (1); take private / extra lessons (2); ask teacher (1); decompose problems (1); ask online (1); (in the context of video-game programming) find and remember errors early to avoid them later (1).

OQ2 Categories: make students have fun (3); repeat things often (3); link to student interest/passion (3); limit boring explanation/theory (2); encourage group work (2); involve students in logic reasoning (2); deliver interactive lessons (1); avoid numeric grades (1); propose creative exercises (1); link subject to real life (1); teach value of effort and training (1); personalize teaching (1).

OQ3 Categories: program on your own at home constantly (9); watch online video tutorials/examples (6); constantly study (5); have objectives/interest/motivation regarding CS (4); believe in yourself/ do not discourage (4); ask questions (4); pay attention during lessons (3); study the programming language (syntax) (3); understand fundamentals (2); take notes (2); not remain behind: concepts builds on previous ones (2); you can do everything with programming (1); study theory (1); there are many ways to solve a problem (1); reflect on errors (1); understand instead of learning by hearth (1); break down the problem in sub-problems (1); read compiler error messages (1); try to engage yourself by programming video games (1).

5 DISCUSSION, CONCLUSIONS AND FURTHER WORK

We found (RQ1) no statistical evidence that studying (or not studying) CS for one year automatically fosters growth mindset in high school students. Moreover, we did not find significant differences between males and females, students with or without previous programming experiences, CS or non-CS students.

We see these results neither as surprising nor as negative, since they are in line with educational research, stating that transfer is difficult between distant domains and does not happen automatically [2]. Our results support warnings about enthusiastic claims around CS fostering GM.

We still think some characteristics of CS can help to develop a GM, but only if teaching interventions are explicitly designed to do so (and if they leverage, for example, on the creative and iterative nature of CS, as we have successfully shown in a preliminary study [20]).

CSEd research should also focus on the opposite implication: CS is a challenging subject, therefore a growth mindset can be particularly important to succeed in it [18, 22].

Table 4: Independent samples tests

	M		F		t/Wilcocxon			
n	56		10					
	m	sd	m	sd	t	W	p	d
GM pre	3.99	1.27	4.17	1.36	-0.39		0.70	
GM pos	3.70	1.40	4.40	1.12		194	0.12	
GM diff	-0.30	1.40	0.23	1.37	-1.11		0.27	
CSM pre	4.36	0.96	4.47	1.36		236	0.43	
CSM pos	4.06	1.04	3.70	1.33	0.96		0.34	
CSM diff	-0.30	0.96	-0.78	0.88		359	0.16	
	EXP		NO EXP		t/Wilcocxon			
n	30		36					
	m	sd	m	sd	t	W	p	d
GM pre	4.07	1.21	3.98	1.35		512	0.72	
GM pos	3.94	1.20	3.69	1.51	-0.76		0.45	
GM diff	-0.12	1.36	-0.30	1.44	-0.50		0.62	
CSM pre	4.28	0.99	4.46	1.04		593	0.50	
CSM pos	4.21	1.02	3.83	1.13	-1.41		0.16	
CSM diff	-0.08	0.91	-0.63	0.94		395	0.06	0.6
	CS		NO CS		t/Wilcocxon			
n	37		29					
	m	sd	m	sd	t	W	p	d
GM pre	4.08	1.33	3.94	1.23		496	0.60	
GM pos	3.78	1.53	3.83	1.17		544	0.93	
GM diff	-0.30	1.59	-0.12	1.12	0.52		0.60	
CSM pre	4.38	0.98	4.38	1.07	0.00		1.00	
CSM pos	4.14	1.08	3.84	1.10	-1.11		0.27	
CSM diff	-0.24	0.98	-0.54	0.92		371	0.03*	0.3

Table 5: Correlation between grades and mindsets

Grades					
GM	r	sig.	CSM	r	sig.
pre	0.04	0.80	pre	0.07	0.70
post	0.27	0.11	post	0.16	0.33
diff	0.22	0.19	diff	0.11	0.50

The intervention where we explicitly taught about Growth Mindset in one of the CS classes (INF1) was not effective (RQ2): the average mindset of INF1 students decreased (although not significantly) from 4.02 out of 6 to 3.33 out of 6. One of our possible explanations is the fact that the initial discussion during the lesson really engaged students and took much more time than scheduled, forcing the researcher to rush through the subsequent points. Furthermore, a shift towards a more fixed mindset was also found after a similar intervention in CS1 courses [30]. Finally, the efficacy of this kind of intervention is debated [31].

However, INF1 students, after receiving the intervention, gave to hypothetical colleagues coherent GM suggestions (RQ3) to overcome difficulties (put effort, ask questions, practice regularly and so on). Surprisingly, when asked what a teacher should do to stimulate a growth mindset, almost no one answered with typical GM teaching suggestions (e.g., teach explicitly about brain growth, praise effort): the vast majority advocated more creativity, engagement, fun, connection with the real world or student passions. This is interesting because in other fields, for example, engineering, the suggested approaches proved to be effective: during the first year of University, students usually move towards a fixed mindset. However, introducing open-ended, socially relevant engineering design projects into the curriculum helps to lessen or eliminate the shift toward fixed mindset [26]. Our preliminary results on Primary Education students seem to agree with that [20].

We found (RQ1) a statistically significant decrease in "Computer Science mindset" only in students not studying CS. The decrease was higher for girls and students without any previous programming experience. This is not desirable: if we think CS has a universal social value, reinforcing stereotypes about a "geek gene" [1, 24] will be harmful and lead to a problem similar to those many students are experiencing in Math, as international tests reveal. This adds to the evidence on the importance of introducing CS principles for all K-12 students.

Identification with CS decreased (RQ1) in CS students, confirming what was already observed for Math [37]. Its (small) increase in non-CS students is worth further analysis.

This research was conducted in a high school in Italy, with 16-year-old students. To generalize the results, it should be reproduced with different samples (different kind of schools, different geographic areas, different age and previous experience, different teaching methodologies, and so on). Items on CS beliefs were translated into Italian and adapted from recent Math research: thus, we need validation studies and better calibration to assess CS beliefs effectively.

ACKNOWLEDGMENTS

Thanks to the students involved and to their teachers, especially to N.A. for believing in the study. Thanks to Alle Serra for creating the link with the school. Thanks to my supervisor Simone Martini for the support; to Tim Bell for hosting me; and to both for providing useful suggestions.

REFERENCES

[1] Alireza Ahadi and Raymond Lister. 2013. Geek genes, prior knowledge, stumbling points and learning edge momentum. *Proceedings of the ninth annual international ACM conference on International computing education research - ICER '13* (2013). https://doi.org/10.1145/2493394.2493416

[2] Susan A. Ambrose, Michael W. Bridges, Michele DiPietro, Marsha C. Lovett, and Marie K. Norman. 2010. *How learning works: Seven research-based principles for smart teaching.* John Wiley & Sons.

[3] Miles Berry. 2015. Teaching computing. Retrieved January 20, 2019 from http://milesberry.net/2015/02/teaching-computing/

[4] Lisa S. Blackwell, Kali H. Trzesniewski, and Carol S. Dweck. 2007. Implicit Theories of Intelligence Predict Achievement Across an Adolescent Transition: A Longitudinal Study and an Intervention. *Child Development* 78, 1 (Jan 2007), 246–263. https://doi.org/10.1111/j.1467-8624.2007.00995.x

[5] Jo Boaler. 2013. Ability and mathematics: the mindset revolution that is reshaping education. In *Forum*, Vol. 55. Symposium Journals, 143–152.

[6] Quintin Cutts, Emily Cutts, Stephen Draper, Patrick O'Donnell, and Peter Saffrey. 2010. Manipulating mindset to positively influence introductory programming performance. *Proceedings of the 41st ACM technical symposium on Computer science education - SIGCSE '10* (2010). https://doi.org/10.1145/1734263.1734409

[7] Carol S. Dweck. 2000. *Self-theories: Their role in motivation, personality, and development.* Psychology Press.

[8] Carol S. Dweck. 2003. *Teorie del Sé. Intelligenza, motivazione, personalità e sviluppo.* Erickson.

[9] Carol S. Dweck. 2008. Mindsets and Math/Science Achievement. *The Opportunity Equation* (2008).

[10] Carol S. Dweck. 2017. *Mindset (Updated Edition).* Robinson.

[11] Abraham E. Flanigan, Markeya S. Peteranetz, Duane F. Shell, and Leen-Kiat Soh. 2015. Exploring Changes in Computer Science Students' Implicit Theories of Intelligence Across the Semester. *Proceedings of the eleventh annual International Conference on International Computing Education Research - ICER '15* (2015). https://doi.org/10.1145/2787622.2787722

[12] Catherine Good, Aneeta Rattan, and Carol S. Dweck. 2012. Why do women opt out? Sense of belonging and women's representation in mathematics. *Journal of Personality and Social Psychology* 102, 4 (2012), 700–717. https://doi.org/10.1037/a0026659

[13] Karin Schreier Hallett. 2016. Building Growth Mindset Through Coding. Retrieved January 20, 2019 from https://liquidliteracy.com/2016/12/28/building-growth-mindset-through-coding/

[14] Nigel Holmes. 2016. Two mindsets. Retrieved January 20, 2019 from http://www.nigelholmes.com/site/wp-content/uploads/2016/09/two$_{m}$indsets.png

[15] Antti-Juhani Kaijanaho and Ville Tirronen. 2018. Fixed Versus Growth Mindset Does Not Seem to Matter Much: A Prospective Observational Study in Two Late Bachelor Level Computer Science Courses. In *Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18).* ACM, New York, NY, USA, 11–20. https://doi.org/10.1145/3230977.3230982

[16] Khan Academy and PERTS. 2014. Growth Mindset Lesson Plan. Retrieved January 20, 2019 from https://cdn.kastatic.org/KA-share/Toolkit-photos/FINAL+Growth+Mindset+Lesson+Plan.pdf

[17] David Klahr and Sharon McCoy Carver. 1988. Cognitive objectives in a LOGO debugging curriculum: Instruction, learning, and transfer. *Cognitive Psychology* 20, 3 (1988), 362 – 404. https://doi.org/10.1016/0010-0285(88)90004-7

[18] Colleen M. Lewis. 2017. *Good (and Bad) Reasons to Teach All Students Computer Science.* Springer International Publishing, Cham, 15–34. https://doi.org/10.1007/978-3-319-54226-3$_{2}$

[19] Michael Lodi. 2017. Growth Mindset in Computational Thinking Teaching and Teacher Training. In *Proceedings of the 2017 ACM Conference on International Computing Education Research (ICER '17).* ACM, New York, NY, USA, 281–282. https://doi.org/10.1145/3105726.3105736

[20] Michael Lodi. 2018. Can Creative Computing Foster Growth Mindset?. In *Joint Proceedings of the 1st Co-Creation in the Design, Development and Implementation of Technology-Enhanced Learning workshop (CC-TEL 2018) and Systems of Assessments for Computational Thinking Learning workshop (TACKLE 2018) co-located with 13th European Conference on Technology Enhanced Learning (ECTEL 2018), Leeds, United Kingdom, September 3rd, 2018. (CEUR Workshop Proceedings)*, Alicja Piotrkowicz, Rosie Dent-Spargo, Sebastian Dennerlein, István Koren, Panagiotis Antoniou, Paul Bailey, Tamsin Treasure-Jones, Ilenia Fronza, and Claus Pahl (Eds.), Vol. 2190. CEUR-WS.org. http://ceur-ws.org/Vol-2190/TACKLE$_{2}$018$_{p}$aper$_{3}$.pdf

[21] Michael Lodi. 2019. Appendix to "Does Studying CS Automatically Foster a Growth Mindset?". Retrieved April 6, 2019 from https://lodi.ml/appendix-mindset-iticse19

[22] Laurie Murphy and Lynda Thomas. 2008. Dangers of a fixed mindset. *ACM SIGCSE Bulletin* 40, 3 (Aug 2008), 271. https://doi.org/10.1145/1597849.1384344

[23] Jum C Nunnally. 1978. *Psychometric Theory: 2d Ed.* McGraw-Hill.

[24] Elizabeth Patitsas, Jesse Berlin, Michelle Craig, and Steve Easterbrook. 2016. Evidence That Computer Science Grades Are Not Bimodal. In *Proceedings of the 2016 ACM Conference on International Computing Education Research (ICER '16).* ACM, New York, NY, USA, 113–121. https://doi.org/10.1145/2960310.2960312

[25] R Core Team. 2013. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/

[26] Kenneth J. Reid and Daniel M. Ferguson. 2014. Do design experiences in engineering build a growth mindset in students?. In *2014 IEEE Integrated STEM Education Conference.* 1–5. https://doi.org/10.1109/ISECon.2014.6891046

[27] Ronny Scherer. 2016. Learning from the Past–The Need for Empirical Evidence on the Transfer Effects of Computer Programming Skills. *Frontiers in Psychology* 7 (2016), 1390. https://doi.org/10.3389/fpsyg.2016.01390

[28] Michael J. Scott and Gheorghita Ghinea. 2014. On the Domain-Specificity of Mindsets: The Relationship Between Aptitude Beliefs and Programming Practice. *IEEE Transactions on Education* 57, 3 (Aug 2014), 169–174. https://doi.org/10.1109/TE.2013.2288700

[29] Sentis. 2012. Neuroplasticity. Retrieved January 20, 2019 from https://www.youtube.com/watch?v=ELpfYCZa87g

[30] Beth Simon, Brian Hanks, Laurie Murphy, Sue Fitzgerald, Renée McCauley, Lynda Thomas, and Carol Zander. 2008. Saying isn't necessarily believing. *Proceeding of the fourth international workshop on Computing education research - ICER '08* (2008). https://doi.org/10.1145/1404520.1404537

[31] Victoria F. Sisk, Alexander P. Burgoyne, Jingze Sun, Jennifer L. Butler, and Brooke N. Macnamara. 2018. To What Extent and Under Which Circumstances Are Growth Mind-Sets Important to Academic Achievement? Two Meta-Analyses. *Psychological Science* 29, 4 (2018), 549–571. https://doi.org/10.1177/0956797617739704 arXiv:https://doi.org/10.1177/0956797617739704 PMID: 29505339.

[32] Kelly Smith. 2016. Why code club is the best way to develop a growth mindset. Retrieved January 20, 2019 from https://medium.com/coding-at-the-library/why-code-club-is-the-best-way-to-develop-a-growth-mindset-2ac354ab7322

[33] Katherine L. Sun. 2015. *There's no limit: mathematics teaching for a growth mindset.* Ph.D. Dissertation. Graduate School of Education, Stanford University. http://purl.stanford.edu/xf479cc2194

[34] Arjun Venkataswamy. 2016. How Learning to Code Can Help Develop a Growth Mindset. Retrieved January 20, 2019 from https://www.digitaladventures.com/news/2016/5/2/how-learning-to-code-can-help-develop-a-growth-mindset

[35] Kristin Villanueva. 2018. How Having a Growth Mindset Can Help You Learn to Code. Retrieved January 20, 2019 from http://blog.mindsetworks.com/entry/how-having-a-growth-mindset-can-help-you-learn-to-code

[36] Kristin Villanueva. 2018. How Learning to Code Can Help You Develop a Growth Mindset. Retrieved January 20, 2019 from http://blog.mindsetworks.com/entry/how-learning-to-code-can-help-you-develop-a-growth-mindset

[37] Allan Wigfield, Jacquelynne S. Eccles, Douglas Mac Iver, David A. Reuman, and Carol Midgley. 1991. Transitions during early adolescence: Changes in children's domain-specific self-perceptions and general self-esteem across the transition to junior high school. *Developmental psychology* 27, 4 (1991), 552.