Impact of Hour of Code: A Five-Year Study

Completed Research

Jie Du
Grand Valley State University
dujie@gvsu.edu

Hayden Wimmer
Georgia Southern University
hwimmer@georgiasouthern.edu

Abstract

This research aims to understand the impact of Hour of Code in consecutive 5 year since it was first launched in 2013. College students in an introductory computing course were surveyed about their attitude toward programming and their programming knowledge. We found evidence of positive impact of Hour of Code on student attitude toward programming as well as their understanding of programming knowledge. This study provides a foundation upon which further investigation into gender differences in computing can be based as well as improving attitude, knowledge, and interest in computer programming.

Keywords

Hour of Code, computer science, programming, attitude, gender.

Introduction

It is well-accepted that there is a shortage of computing professionals entering the workforce which will culminate in a myriad of challenges in the future. Similarly, it is also well-accepted that there is a lack of diversity, both in terms of race and gender, in the computing profession. We posit that the aforesaid challenges are not mutually exclusive. Improving the diversity of the workforce has the potential to alleviate the shortage of computing professionals as currently less than half the US population (females) is represented in the computing workforce. One such effort to improve the numbers and diversity in the computing workforce comes from Code.org's Hour of Code (HoC) initiative. HoC seeks to have an international event where students spend 1 hour writing computer code. This effort has the ability to improve students' attitudes toward computing, self-efficacy, intention to pursue computing, and many other related constructs. By increasing exposure to computer programming and thereby improving programming knowledge and attitude towards computing, it is possible to recruit more students into computing majors and cognate careers. As it stands, especially in the U.S., computing is sporadic in secondary education and oftentimes not required as part of middle and high school curriculums. Due to this, students' first exposure is often at the collegiate level. By introducing computing, specifically computer programming, during entry level courses at the college level, we expand exposure to the computing discipline. Moreover, using core courses which are required of all majors expands the exposure and thereby facilitates recruiting women and diversity into computing and cognate careers.

As one of the most popular initiatives to promote computer science (CS), Hour of Code serves as an incubator to recruit more students to try computer science. This study attempts to gain insight on such impact by surveying students in an introductory computing course about their attitude toward programming as well as their knowledge on programming in 5 consecutive years from 2014 to 2018.

Background

What is Hour of Code?

Code.org was launched as a nonprofit organization in 2013 with aim to promoting computer science. In an interview by Lawrence Snyder (2013), Hedi Parovi, the founder and CEO of Code.org said that "my effort is to turbocharge computer science education." In order to realize their vision that every student in every school should have the opportunity to learn computer science, Code.org hosts a curriculum of over 100, one-hour-long, computer science activities, which is known as Hour of Code.

Bringing computer programming into the mainstream dialogue and raising national awareness about computer programming is one of the missions of Code.org. Bringing computer science into every American public school across the nation, by district, city, until CS becomes a main part of US education is discussed by (Partovi, 2014). To address this, Partovi (2014) launched a campaign called 'Hour of Code" that had more than 3.5 million students try a programming example online. HoC tutorials are online, web-based, and game-based. HoC aims to demystify coding and teach students computer science basics in a fun way. Students learn coding by playing a game, such as Minecraft. More recently, Code.org teamed up with Disney to offer programming lessons that include wildly popular characters from "Frozen" designed to attract more girls to participate (Layton, 2017). To date, over 735 million students have tried the Hour of Code and among them 49% are females (Code.org, n.d.). According to Wilson (2015), there is a need to have young children get more involved in computer science and programming at an early age. Code.org established a partnership with 60 school districts serving nearly 8 million students. Similarly, Code.org is holding workshops for teachers with 3200 teachers trained at 200 workshops. Moreover, 90,000 classrooms are using Code.org in some form in their teaching practices.

It is well accepted that there is a lack of diversity in the field of computing (Wilson, 2014). To address the issue of diversity in the technology industry, Code.org may be the key. Hour of Code had approximately 50% females complete the challenge in 2014. The HoC inspires and creates opportunities for boys and girls of all backgrounds to try computer science. Code.org is tackling the diversity problem in computing by implementing computer science in elementary and middle schools before stereotypes surrounding computer science kick in. Additionally, Code.org is actively running campaigns worldwide and having tutorials translated into over 30 languages. Code.org has also been established in cities in Europe and Brazil to help grow the movement internationally. Imploring tech companies to support Hour of Code events at offices, local schools, and donating to campaigns to enable students participate globally is another strategy. Results seem to be positive and show that 37 percent women are enrolled for pilot courses and 60 percent are African American or Hispanic. Similarly, the demographic breakdown of students taking computer science exam in high school was 20 percent female and 17 percent African American or Hispanic (Wilson, 2014).

Impact of Hour of Code

Code.org was mainly referred to as an advocacy effort for computer education when it was first launched in 2013 (Meeker, 2014; Mueller, 2013; Verkroost, 2013). Recently, researchers investigated the impact of Hour of Code that goes beyond an advocacy effort. Kalelioğlu (2015) saw a need to improve classroom teaching and reflective thinking skills toward problem solving. To address the need, a new method to teach programming to K-12 students via Code.org is demonstrated. Kalelioğlu sought to investigate if there is a gender difference in terms of students' reflective thinking skills on problem solving. To determine the answer to this question, a triangular study was conducted with 32 primary students, a qualitative and quantitative study on students' performance that includes pre and posttest comparison quasi design, creating focus groups and an analysis of reflection paper from the IT teacher as conducted. T-tests were also carried out on analysis. Results showed that teaching programming to primary school students didn't cause any difference in reflective thinking skills toward problem solving. But there is a little increase in the average of female students' reflective thinking skills over males. Students also developed a positive attitude toward programming (Kalelioğlu, 2015).

Phillips and Brooks (2017) studied student attitude and self-efficacy of computer science via an Hour of Code activity. A survey was conducted on students before and after an Hour of Code activity and two-part questions were asked, one part on attitude and the second on self-efficacy, and additional questions on prior experience with computer science. Results indicated that student attitudes and self-efficacy with computer science increased after an Hour of Code activity. This shows that having the right activity and easy exposure to computer science is an important way to involve more students and more females in computer science.

While substantial research exists on the Hour of Code curriculum for K-12 computer science, there appears to be very little research on its impact on college level computer science curriculum. Du, Wimmer, and Rada (2016) examined the lack of student interest in programming and sought to determine the effect of Hour of Code activity in students' programming skills. 116 college students participated in a tutorial as an Hour of Code initiative and were surveyed. The survey included questions on programming, attitude toward programming, and employed a pre/post-test method. Results show that the Hour of Code tutorial had a

positive impact on student attitude toward programming but did not significantly improve students' programming skills.

It has been five years since the Hour of Code initiatives were first launched in November 2013. What is the impact of Hour of Code on promoting CS education in colleges in the past 5 years? To address this question, we set out to investigate the impact of Hour of Code on college students. To the best of our knowledge, this is the first study that investigates the impact of Hour of Code in 5 consecutive years since it was first launched. In this work we focus on improving attitude and coding knowledge regardless of any confounding factors. Factors such as entertainment value, prior education, or a number of other factors may contribute to improving attitude as part of the tutorial. Finally, we advance our formal hypotheses as:

- H1: The Hour of Code tutorials can positively change students' attitude toward programming.
- H2: The Hour of Code tutorials can improve students' coding knowledge.

Methodology

This study aims to explore the impact of the Hour of Code initiatives on college students. In order to study the impact of Hour of Code, each year for 5 years we followed a pre/post-survey technique on college students to measure improvement in coding as well as improvement in students' perceptions and attitude toward computer programming. We first analyze the data from each year separately and graph each years' results. Additionally, we analyze the dataset as a whole combining all years to determine any improvement in coding ability or perceptions and attitude of computer programming.

Data Collection

Data were collected from 2014 to 2018 from college students enrolled in an introductory computing course which is required of all College of Business students as well as other majors. This study was conducted when software programming was discussed mid-semester. After removing incomplete responses, the data set yields 219 usable responses (44 in 2014, 45 in 2015, 47 in 2016, 40 in 2017, and 43 in 2018). An electronic Likert-scale questionnaire was implemented to survey the participants about their attitude and skills toward programming. The survey contains 14 questions based on (Du et al., 2016). This study was conducted in a 50-min class time and had three steps: 1) The participants completed the pre-survey that contains Q1 to Q11 plus Q14 before taking the Hour of Code tutorial, 2) The participants completed the tutorial "Write Your First Computer Program" from the category of "Tutorial for Beginners" at Code.org, and 3) the participants took the post survey that contains Q1, Q6 to Q13 after taking the Hour of Code tutorial.

Data Processing and Analysis

Participants' responses to the pre- and post-surveys were matched using a PIN number created by each participant. The data collected from the survey were first processed to numerical values according to the following rules: 1) The responses were recorded as 1 or 2, corresponding to yes or no, respectively, 2) The responses were recorded as 1 to 4, corresponding to less than 1 year to 5 years, respectively, 3) The responses were recorded as 1 to 5, corresponding to strongly disagree to strongly agree, respectively, 4) The responses were recorded as 1 to 4, corresponding to not at all likely to very likely, respectively, and 5) The responses were recorded as 1 or 0, depending on whether it matches the correct answer. Then the processed data were analyzed in two steps: 1) analyzing the 5 yearly data sets by conducting paired sample *t*-tests on the five individual yearly data sets to test the two hypotheses, and 2) analyzing the five-year data set as a whole and performing a 5-year trend analysis on student attitude toward programming and their coding skills which includes studying the means and correlations.

Results

Yearly Data Analysis - 2014 Data

A paired *t*-test was conducted on the 2014 data set to detect the difference on responses between pre- and post-surveys (see Table 1). The results show that the participants' responses to the attitude questions (Q7

and Q8) are significantly different in the pre- and post-surveys. Therefore, H1 was supported. The participants' attitude toward programming was positively impacted after taking the Hour of Code tutorial. Figures 1 and 2 show the response distribution on the attitude questions. There is no significant difference on the three coding knowledge questions (Q9, Q10, and Q11). Thus, H2 was not supported. The participants' knowledge on programming was not significantly improved after taking the Hour of Code tutorial.

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	Interval of the Lower	Upper			
Pair 1	Q7_Pre - Q7_Post	-0.727	0.788	0.119	-0.967	-0.488	-6.119	43	0.000
Pair 2	Q8_Pre - Q8_Post	-0.409	0.622	0.094	-0.598	-0.220	-4.363	43	0.000
Pair 3	Q9_Pre - Q9_Post	-0.023	0.549	0.083	-0.190	0.144	-0.274	43	0.785
Pair 4	Q10_Pre - Q10_Post	0.045	0.569	0.086	-0.127	0.218	0.530	43	0.599
Pair 5	Q11_Pre - Q11_Post	0.000	0.374	0.056	-0.114	0.114	0.000	43	1.000

Table 1. T-test on 2014 Data

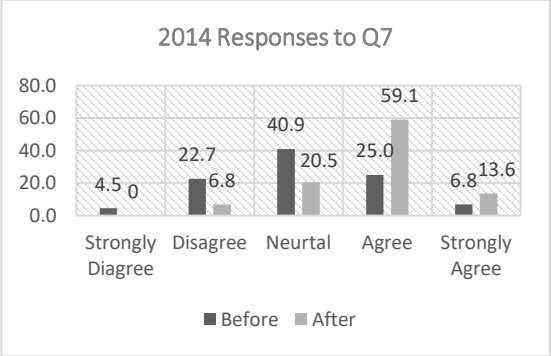

Figure 1. 2014 Responses to Q7

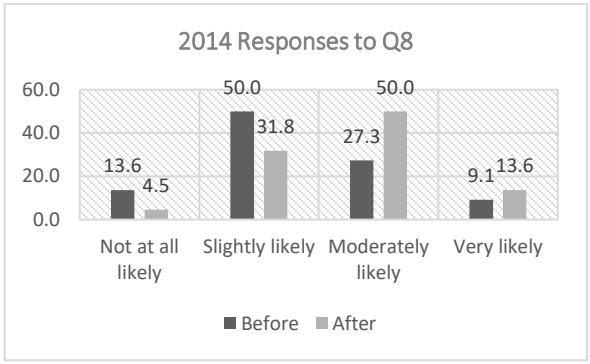

Figure 2. 2014 Responses to Q8

Yearly Data Analysis - 2015 Data

A paired *t*-test was conducted on the 2015 data set to detect the difference on responses between pre- and post-surveys (see Table 2). The results show that the participants' responses to the attitude questions (Q7 and Q8) are significantly different in the pre- and post-surveys. 31.1% of the participants agreed that everyone should learn programming before taking any hour of code tutorial. This number doubled after the participants took an Hour of Code tutorial (see Figure 3). 40% of the participants are likely to take a programming course after taking the Hour of Code tutorial compared to 26.6% in the pre-survey (see Figure 4). Therefore, H1 was supported. The participants' attitude toward programming was positively impacted after taking the Hour of Code tutorial. There is no significant difference on the three coding knowledge questions (Q9, Q10, and Q11). Thus, H2 was not supported.

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	Interval of the Lower	Upper			
Pair 1	Q7_Pre - Q7_Post	-.800	1.014	.151	-1.105	-.495	-5.295	44	.000
Pair 2	Q8_Pre - Q8_Post	-.422	.621	.093	-.609	-.236	-4.560	44	.000
Pair 3	Q9_Pre - Q9_Post	0.000	.603	.090	-.181	.181	0.000	44	1.000
Pair 4	Q10_Pre - Q10_Post	-.044	.520	.078	-.201	.112	-.573	44	.570
Pair 5	Q11_Pre - Q11_Post	-.044	.298	.044	-.134	.045	-1.000	44	.323

Table 2. T-test on 2015 Data

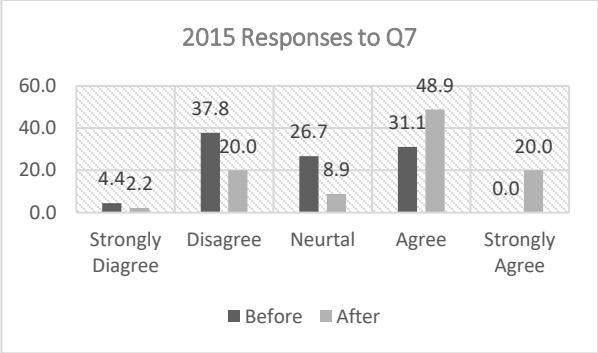

Figure 3. 2015 Responses to Q7

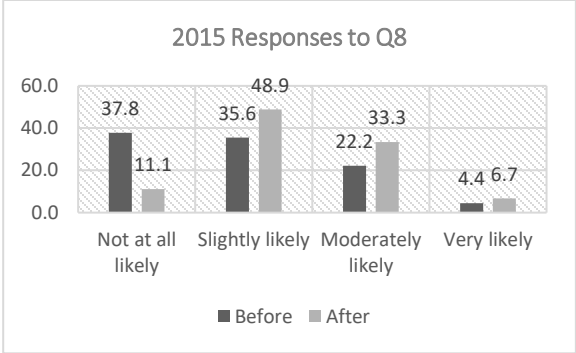

Figure 4. 2015 Responses to Q8

Yearly Data Analysis - 2016 Data

Another paired sample *t*-test was conducted on the 2016 data. The *t*-test results show that the participants' responses to the attitude questions are significantly different in the pre- and post-surveys (see Table 3). More participants are having a positive attitude toward programming (see Figure 5) and are likely to take a programming course after taking the Hour of Code tutorial (see Figure 6). Therefore, H1 was supported. There is no significant difference on the three coding knowledge questions. Thus, H2 was not supported.

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	Interval of the Lower	Upper			
Pair 1	Q7_Pre - Q7_Post	-.298	1.140	.166	-.633	.037	-1.791	46	.080
Pair 2	Q8_Pre - Q8_Post	-.234	.520	.076	-.387	-.081	-3.087	46	.003
Pair 3	Q9_Pre - Q9_Post	-.043	.359	.052	-.148	.063	-.814	46	.420
Pair 4	Q10_Pre - Q10_Post	-.149	.589	.086	-.322	.024	-1.734	46	.090
Pair 5	Q11_Pre - Q11_Post	0.000	.466	.068	-.137	.137	0.000	46	1.000

Table 3. T-test on 2016 Data

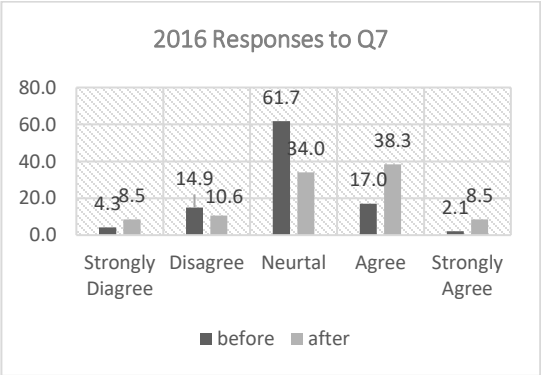

Figure 5. 2016 Responses to Q7

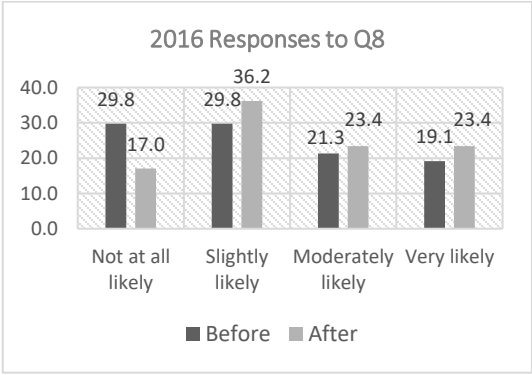

Figure 6. 2016 Responses to Q8

Yearly Data Analysis - 2017 Data

A paired *t*-test was conducted on the 2017 data set to detect the difference on responses between pre- and post-surveys (see Table 4). The results show that the participants' responses to the attitude questions (Q7 and Q8) are significantly different in the pre- and post-surveys. 35% of the participants agreed that everyone should learn programming before taking any Hour of Code tutorial. However, this number increased to

57.5% after the participants took an Hour of Code tutorial (see Figure 7). Also 50% of the participants are likely to take a programming course after taking the Hour of Code tutorial compared to 37.5% in the pre-survey (see Figure 8). Therefore, H1 was supported. There is no significant difference on the three coding knowledge questions. Thus, H2 was not supported.

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	Interval of the Lower	Upper			
Pair 1	Q7_Pre - Q7_Post	-0.300	0.758	0.120	-0.542	-0.058	-2.504	39	0.017
Pair 2	Q8_Pre - Q8_Post	-0.300	0.564	0.089	-0.480	-0.120	-3.365	39	0.002
Pair 3	Q9_Pre - Q9_Post	0.025	0.577	0.091	-0.159	0.209	0.274	39	0.785
Pair 4	Q10_Pre - Q10_Post	-0.050	0.450	0.071	-0.194	0.094	-0.703	39	0.486
Pair 5	Q11_Pre - Q11_Post	-0.125	0.516	0.082	-0.290	0.040	-1.533	39	0.133

Table 4. T-test on 2017 Data

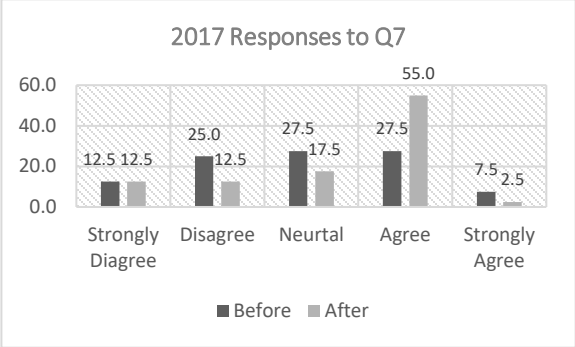

Figure 7. 2017 Responses to Q7

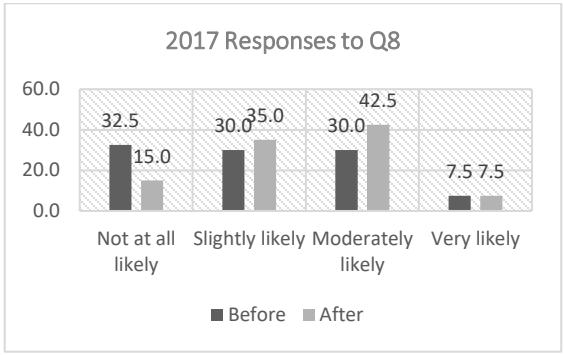

Figure 8. 2017 Responses to Q8

Yearly Data Analysis - 2018 data

A paired t-test was conducted on the 2018 data set to detect the difference on responses between pre- and post-surveys (see Table 5). The results show that the participants' responses to the attitude questions (Q7 and Q8) are significantly different in the pre- and post-surveys. 67.4% of the participants agreed that everyone should learn programming after taking any hour of code tutorial (see Figure 9). Also 55.9% of the participants are likely to take a programming course after taking the hour of code tutorial compared to 41.9% in the pre-survey (see Figure 10). Therefore, H1 was supported. We found that the participants had a significantly improved understanding on the IF structure (Q9) after taking the Hour of Code tutorial. Thus H2 was partially supported.

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	Interval of the Lower	Upper			
Pair 1	Q7_Pre - Q7_Post	-0.465	0.631	0.096	-0.659	-0.271	-4.837	42	0.000
Pair 2	Q8_Pre - Q8_Post	-0.302	0.513	0.078	-0.460	-0.144	-3.862	42	0.000
Pair 3	Q9_Pre - Q9_Post	-0.093	0.294	0.045	-0.183	-0.003	-2.075	42	0.044
Pair 4	Q10_Pre - Q10_Post	-0.070	0.402	0.061	-0.194	0.054	-1.138	42	0.262
Pair 5	Q11_Pre - Q11_Post	0.070	0.457	0.070	-0.071	0.211	1.000	42	0.323

Table 5. T-test on 2018 Data

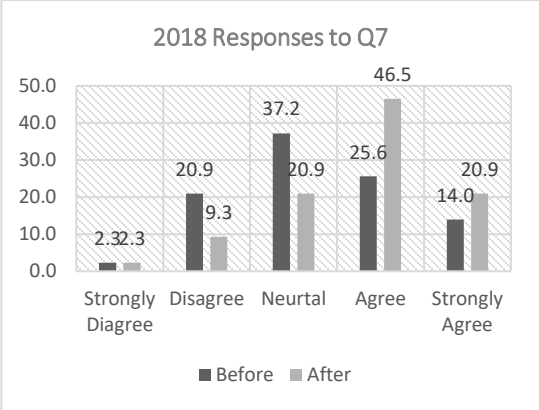

Figure 9. 2018 Responses to Q7

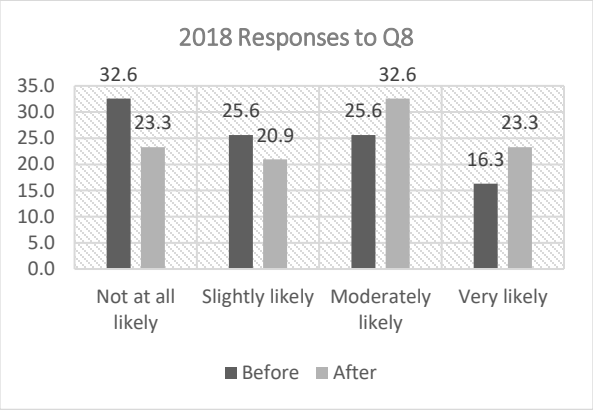

Figure 10. 2018 Responses to Q8

Analysis – Combined Data

Attitude toward Programming

Students' attitude toward programming has been significantly positively impacted after taking the Hour of Code tutorial (see Table 6). The results are consistent with the yearly data analysis results. More participants expressed positive attitude toward programming (see Figure 11) and interest in taking programming courses after completing the hour of code tutorial (see Figure 12).

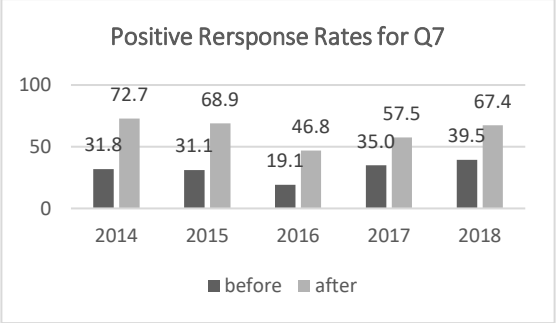

Figure 11. Responses to Q7

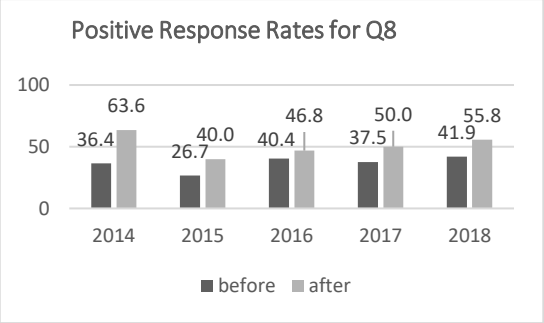

Figure 12. Responses to Q8

		Paired Differences					t	df	Sig. (2-tailed)
		Mean	Std. Deviation	Std. Error Mean	Interval of the Lower	Interval of the Upper			
Pair 1	Q7_Pre - Q7_Post	-0.521	0.910	0.062	-0.642	-0.399	-8.462	218	0.000
Pair 2	Q8_Pre - Q8_Post	-0.169	0.638	0.043	-0.254	-0.084	-3.919	218	0.000
Pair 3	Q9_Pre - Q9_Post	-0.027	0.488	0.033	-0.092	0.038	-0.831	218	0.407
Pair 4	Q10_Pre - Q10_Post	-0.055	0.513	0.035	-0.123	0.014	-1.581	218	0.115
Pair 5	Q11_Pre - Q11_Post	-0.018	0.428	0.029	-0.075	0.039	-0.632	218	0.528

Table 6. T-test for the Whole Dataset

Coding Skills (Accuracy Rate)

We did not find any significant change on the responses to the three questions that examined participants' understanding about programming structure (see Table 6). One possible reason is the responses to these three questions were processed to 1 or 0 depending on whether the answer provided is correct or not. This

processing minimized the differences between each response and may bias the *t*-test result. To detect the changes in students' understanding about coding, we examined their accuracy rate on these three questions. The individual accuracy rate was first calculated on each question. Next a weighted accuracy rate was calculated by summing up the three individual accuracy rates and dividing by three. Figure 13 shows the accuracy rates before and after the participants took the Hour of Code tutorial. The *t*-test result shows that the accuracy rates are significantly different before and after the participants completed the Hour of Code tutorial (t=-2.745, p=0.05). It is interesting to find that the participants obtained a better, average accuracy rate on the three questions after they took the Hour of Code tutorials except for year 2014.

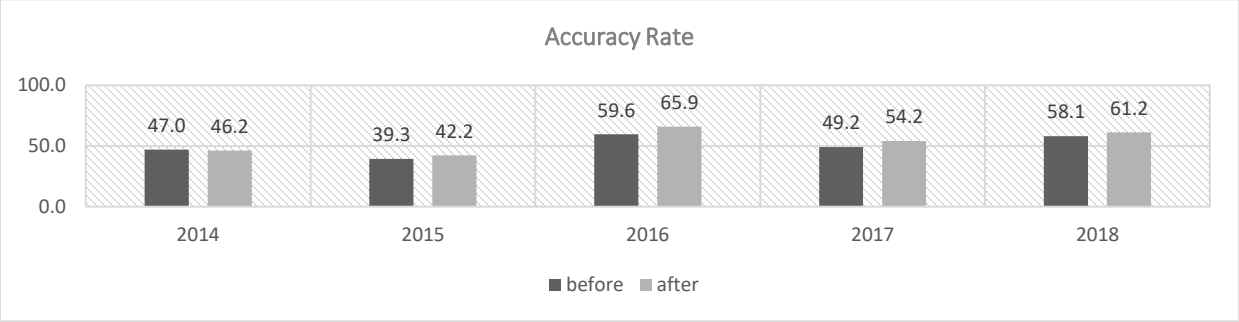

Figure 13. Accuracy Rates in the Five Years

Feedback on Hour of Code

Two questions were only asked in the post survey about the participants' attitude change toward programming (Q13) and whether they enjoy the Hour of Code tutorial (Q12). We compared the responses collected from 2014 to 2018. We found that the majority of the participants enjoyed the Hour of Code tutorials (range from 66% to 84.4%) (see Figure 14). At least 59.6% of the participants believed that the hour of code changed their attitude toward programming positively (see Figure 15).

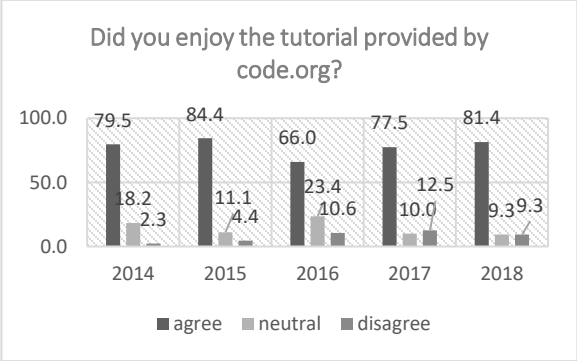

Figure 14. Responses to Q12

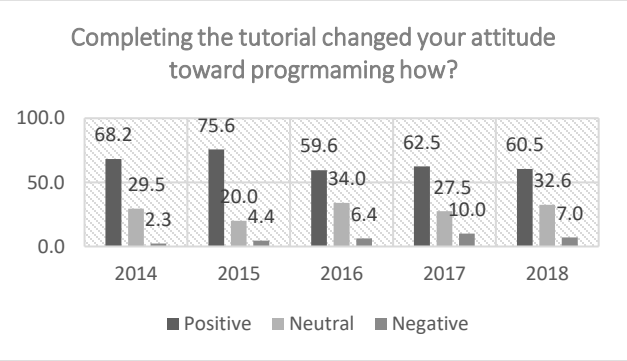

Figure 15. Responses to Q13

Correlations

It is interesting to see how the participants' knowledge about programming correlates with their attitude toward programming. Table 7 shows the correlations between the accuracy rate on Q9, Q10, and Q11, and student attitude toward programming (measured by Q7, Q8, Q12, and Q13). A pattern is detected that the accuracy rate is negatively correlated to student feedback on Hour of Code (Q12 and Q13). For example, the participants from year 2016 expressed the lowest interest (66%) on the tutorial (Q12) compared to other years. However, the same group of the participants obtained the highest accuracy rate (65.9%) compared to other groups in the other years. Another pattern is that the accuracy rate is negatively correlated to the responses to Q7. The data suggest that the more students know about programming, the less they believe that everyone should learn programming. Our data shows no correlation between how likely students are to take a programming course and their accuracy rates.

Correlations	Accuracy Rate
Q7 - do you agree that everyone should learn programming?	-0.74
Q8 - how likely are you to take a programming course?	0.04
Q12 - did you enjoy the hour of code tutorials?	-0.74
Q13 - completing the tutorial changed your attitude toward programming how?	-0.93

Table 7. Correlations between Accuracy Rates and Attitude toward Programming

Discussions

Main Findings

There are three major findings in our study. First, the Hour of Code tutorial can significantly change students' attitude toward programming. The *t*-test results on the five years' data are consistent. There are significant differences on the responses to the two questions related to attitude toward programming (Q7 and Q8) before and after the participants took the Hour of Code tutorial. It is consistent with our analysis on the full data set. Second, we found that the participants obtained a higher average accuracy rate on the questions that examine their understanding on the coding knowledge after they completed the Hour of Code tutorial. Besides demystifying coding, Hour of Code also helps improve students' computing skills. One thing the authors would like to point out is that accuracy rate for the comprehension questions was used in this study to measure changes before and after students completed an hour of code tutorial. In the future, we would like to explore new coding methods based on similarity of options for the comprehension questions. Third, it is interesting to find that the accuracy rate on the coding knowledge questions is negatively correlated to the response to whether the participants enjoy the tutorial. The more the students know about programming, the less the students feel interested in the Hour of Code tutorial and the less the Hour of Code could change their attitude toward programming. We also found that the more students know about programming, the less they believe that everyone should learn programming. This result is contradictory to our expectation. It is possible that the students who know more about programming think that programming means using specific programming languages to write code other than the general problem-solving skills of using a computer and thus conclude that programming is not for everybody. As part of the research, we included some very basic demographic questions including a gender identification question for the purposes of investigating whether or not gender plays a role in attitude toward programming in years 2017 and 2018. The findings based on the gender of the participants have been published in (Du & Wimmer, 2018) and thus were omitted from this paper.

Implications

Our results show a positive impact of Hour of Code on student attitude toward programming. It is suggested that Hour of Code be introduced into classrooms by educators. It is especially appropriate to incorporate Hour of Code into an introduction to computing courses to engage more students in computing. This effort may help address the shortage of computing professionals in the long run. Based on our data, we found that Hour of Code improved students' coding knowledge. So besides using Hour of Code as an advocacy example, educators may consider incorporating Hour of Code tutorials into their curriculum. Over 100 Hour of Code tutorials are provided at Code.org and organized based on learner's experience and grade level. Educators could explore the library of Hour of Code tutorials and utilize the one that best meets their curriculum requirements.

Gender has been shown to play a significant role in education (Byrne, 1978) and computing (Venkatesh, Morris, Davis, & Davis, 2003) research. Stereotypes based on gender widely exist in CS, such as low awareness of female academic competence (Koch, Müller, & Sieverding, 2008). It is well known that a gender gap exits in CS. The percentage of females who took an undergraduate degree in CS in 1970-71 was 14%. This rose to 37% in 1983-84 but gradually declined to 18% in 2010-11 ("Digest of Education Statistics," n.d.). One of the goals of Code.org is to improve diversity in CS. The impact of Hour of Code to close the gender gap will be explored in future work. The primary limitation of this study is the small sample size. The sample provided a good initial foundation for exploring impact of Hour of Code. Future work should consider the incorporation of more participants from a variety of institutions so that a more comprehensive sample can be tested.

Our sample was drawn from a required introductory to computing course required of business majors. The course examines principles of computing and computer software such as word processing and spreadsheets. While some students may have had prior programming knowledge from high-school, the majority of business majors have not been exposed to computer programming. Our work seeks to determine if an HoC tutorial is appropriate for this sample of students (business majors) and, based on our findings, we suggest moving forward an HoC or similar tutorial could be added to introductory computing courses to improve attitude and knowledge of computer programming.

Conclusion

This paper examined the impact of Hour of Code on college students. A survey was administered to undergraduates to collect data about their attitude toward and programming knowledge. This research contributes to a better understanding of the impact of Hour of Code on student attitude toward programming as well as their programming knowledge. Hour of code positively impacts student attitude toward programming and improves their programming knowledge. We suggest that introductory computing courses adopt a HoC, or a similar tutorial, as part of the course in order to have a larger impact and recruit more students into computing majors and cognate careers. Since students come to introductory courses with a diverse background, future research will focus on the current programming knowledge of students and adopting appropriate tutorials given the current skill level. Similarly, future research with a larger sample is needed to determine the impact of Hour of Code and to develop a better understanding of gender differences on computing.

References

Digest of Education Statistics. (n.d.). Retrieved from https://nces.ed.gov/programs/digest/d12/tables/dt12_349.asp

Du, J., & Wimmer, H. (2018). *Hour of Code: A study of gender differences in computing*. Paper presented at the EDSIG Conference on Information Systems & Computing Education, Norfolk, Virginia, US.

Du, J., Wimmer, H., & Rada, R. (2016). "Hour of Code": Can It Change Students' Attitudes toward Programming? *Journal of Information Technology Education: Innovations in Practice, 15*, 52-73.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code. org. *Computers in Human Behavior, 52*, 200-210.

Koch, S. C., Müller, S., & Sieverding, M. (2008). Women and computers. Effects of stereotype threat on attribution of failure. *Comput. Educ., 51*(4), 1795-1803. doi:10.1016/j.compedu.2008.05.007

Layton, L. (2017). Hour of code to feature 'frozen' characters. Retrieved from http://search.proquest.com.ezproxy.gvsu.edu/docview/1626770226?accountid=39473

Meeker, P. H. (2014). *Inspiring a love of computer science through the education of our youth (abstract only)*. Paper presented at the the 45th ACM technical symposium on Computer science education, Atlanta, Georgia, USA.

Mueller, R. (2013). *Attracting Females into ICT in Canada*. Paper presented at the the Information and Communications Technology (ICT) Talent Workshop, Ottawa, Canada.

Partovi, H. (2014). *Transforming US education with computer science*. Paper presented at the Proceedings of the 45th ACM technical symposium on Computer science education.

Phillips, R., & Brooks, B. (2017). *The Hour of Code: Impact on Attitudes towards and Self-Efficacy with Computer Science*. Retrieved from code.org: https://code.org/files/HourOfCodeImpactStudy_Jan2017.pdf

Snyder, L. (2013). An interview with Hadi Partovi. *Commun. ACM, 56*(9), 41-45. doi:10.1145/2500133

Verkroost, Y. (2013). Seriousify and prettify our educational system! ABOUT THE USE OF (SERIOUS) GAMES AND AR IN EDUCATION. *Bachelor Project VU University Amsterdam*. Retrieved from http://www.yordiverkroost.nl/vu/paper.pdf

Wilson, C. (2014). Hour of code: we can solve the diversity problem in computer science. *ACM Inroads, 5*(4), 22-22.

Wilson, C. (2015). Hour of code---a record year for computer science. *ACM Inroads, 6*(1), 22-22.