2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #34259

Using Programming Concept Inventory Assessments: Findings in a First-year Engineering Course

Dr. Krista M. Kecskemety, Ohio State University

Krista Kecskemety is an Assistant Professor of Practice in the Department of Engineering Education at The Ohio State University. Krista received her B.S. in Aerospace Engineering at The Ohio State University in 2006 and received her M.S. from Ohio State in 2007. In 2012, Krista completed her Ph.D. in Aerospace Engineering at Ohio State. Her engineering education research interests include investigating first-year engineering student experiences, faculty experiences, and the connection between the two.

Ada Barach, Ohio State University

Ada recently graduated from The Ohio State University with a B.S. in Computer Science and Engineering. Her undergraduate research was in coding education for first-year students. Ada is currently pursuing a PhD in theoretical computer science at Ohio State.

Connor Jenkins, Ohio State University

Connor Jenkins is currently an undergraduate student pursuing a B.S. in Electrical and Computer Engineering at The Ohio State University. His engineering education research interests include first-year engineering, teaching assistant programs, and technical communication education methods.

Ms. Serendipity S. Gunawardena, Ohio State University

Sery is an undergraduate researcher. She is pursuing a Computer Science & Engineering degree with a Psychology minor. She is from Athens, Ohio and currently resides in Dayton, Ohio. She is a Teaching Assistant for the Honors Fundamentals of Engineering Program and enjoys tutoring younger students. Outside of class, Sery likes calligraphy and playing the piano.

Using Programming Concept Inventory Assessments: Findings in a First-Year Engineering Course

Abstract

This complete research paper examines the use of a programming concept inventory assessment in a first-year engineering course. At The Ohio State University, the first-year engineering program focuses on teaching introductory computer programming skills through MATLAB and C/C++. In recent years, this program has undergone curriculum changes which resulted in a desire to measure impacts on student learning. This led to a need for a validated assessment tool like a concept inventory which is used to assess students' conceptual understandings and misunderstandings. The use of concept inventories is common in STEM fields, specifically physics, and has been gaining popularity over the last 30 years. An existing language-independent programming concept inventory, SCS1, was replicated to create a MATLAB specific version (MCS1) in 2019-2020. Both the language-independent and MATLAB-specific assessment were given to students in autumn 2019 at this university. Because this assessment is given in a course with students with a wide range of programming experiences and different demographics, it was helpful to investigate the student results to see what differences exist in this population. This paper examines the demographic and prior programming experience information collected during the assessment, focusing on prior high school computer science experience, self-identified programming skill, gender, and honors vs. standard courses. Using independent samples t-tests, it was found that students who took a computer science course in high school were more confident in their skills but did not earn significantly higher scores on the assessments. While the average self-reported skill level for women was lower than that of men, there was no significant difference in assessment scores. Finally, it was also found that honors students performed significantly better on the assessments than standard students, however those differences are likely related to the extra programming instruction that those students received. The goal of this study is to provide insight into the programming skills of first-year engineering students with a variety of prior experience and perceived ability.

Introduction and Background

Concept inventories are validated assessments which test student understanding of broad concepts in a given field[1]. They are a collection of multiple-choice questions[2]. Since the development of the Force Concept Inventory for physics education [3], researchers have sought to create concept inventories for other topics in areas such as chemistry, astronomy, geoscience and others [4]. Though Concept Inventories are common in fields like the sciences, [5, 3, 6], computer programming has relatively few assessments [7].

In 2011, a concept inventory was developed for evaluating understanding of basic programming knowledge, called Foundational Computer Science 1 (FCS1) [8].There are a variety of programming languages used and taught in schools. This can pose a challenge in testing, since it can be difficult to determine if a concept or syntax of the language is being tested [9]. Because of this challenge, language-independent concept inventories for computer programming have been developed [8, 10, 11]. In 2016, FCS1 was replicated by Parker et. al to create the language-independent Second Computer Science 1 (SCS1) assessment. Parker et. al argued that the existence of multiple assessments in a given area would reduce the negative impact of readily available questions and answers (such as those found online) that could be looked up and memorized [11].

In 2020, SCS1 [11], was replicated to produce a MATLAB-specific concept inventory, MCS1 [12]. Both assessments are multiple-choice and take approximately one hour to complete. These concept inventories assess student understanding in arrays, basics, for loops, function parameters and return values, if statements, logical operators, recursion, and while loops [11, 12]. Since these topics align with the first-year engineering curriculum at The Ohio State University, these concept inventories were used to evaluate the impacts of the recent curriculum changes.

The first-year engineering curriculum at this university focuses on teaching introductory programming skills. This program has two course sequences: standard and honors. Within the honors course sequence there are two options: honors and honors advanced. The standard course teaches fundamental programming concepts through MATLAB while the honors course teaches both MATLAB and C/C++. The honors advanced course follows the same curriculum as the honors course, but includes additional coursework and is intended for students who have significant coding experience. The first-year engineering program at The Ohio State University enrolled 1,888 students in autumn 2019 of which 408 (21.61%) students were in the honors courses [13]. At the end of the autumn 2019 semester, both MCS1 and SCS1 were given to 672 of these first-year engineering students.

This paper examines the demographic and perceived ability differences of the participants as well as the correlation of these with the participants' assessment scores. The four areas of interest in this study are gender, prior programming experience, self-reported programming skill, and the course track. By having a concept inventory assessment that correlates with final grades, we can determine if there are any correlations between these student factors and the assessment. Students enter the first-year engineering courses with a wide range of prior programming experience and students also have different self-efficacy when it comes to their programming skills. At the end of the first-year program, it is the intent that students have a similar level of ability when it comes to the basic programming fundamentals that are assessed in this concept inventory. While there are 2 different course tracks, honors and standard, students still should be gathering similar baseline knowledge in concepts assessed in the concept inventory. However, the honors course teaches an additional course-worth of programming knowledge so it may be expected that they would perform better on the assessment. Additionally, it is not intended for this assessment to be biased and therefore examining the demographic differences may assist in determining if a bias exists. The goal of this study is to address the following research question: *Which factors (gender, prior programming experience, self-reported programming skill, and course track) are associated with student performance on the programming concept inventory assessments?* By answering this question we can draw some conclusions about the usefulness of this assessment as well as the impact of our course tracks in developing this baseline level of knowledge for our first-year engineering students.

Methods

Concept Inventory

All first-year engineering (FYE) students from The Ohio State University were contacted with the opportunity to take part in the SCS1 and MCS1 assessment surveys. The survey took place approximately two weeks before the end of the autumn 2019 semester, at which point students should have completed their MATLAB instruction. They were incentivized to participate with extra credit in their respective engineering course if they completed the survey. Students who responded could take the survey at an hour-long in-person testing sessions, each proctored by a researcher. There were 21 sessions offered and it resulted in 672 usable participant responses. Student responses were removed if the student did not attempt large sections of the assessment or if the student completed the assessment too quickly, more than 2 standard deviations ($\sigma = 812.3s$) less than the mean time ($\bar{x} = 2183.7s$).

Upon starting the assessment, each participant was automatically and randomly assigned to either MCS1 or SCS1 by the testing software. The survey software used, Qualtrics, was set to evenly present both assessments in the random generation. After 60 minutes, the participant's testing session terminated and they were directed to the demographic portion of the survey. SCS1 participants were provided with a pseudocode guide in accordance with the original testing conditions of SCS1. Soon after submitting their assessments, participants were emailed with their scores in each concept area. Of the 672 usable participant responses, 336 participants completed the SCS1 assessment and 336 participants completed the MCS1 assessment. In the analysis of the assessment one of the questions, Question 4 of SCS1, included a typo which resulted in multiple correct answers to the question. When this question was removed the mean score for the SCS1 assessment shifted less than 1% [12], therefore it is not expected to impact the results presented here since they are presented as the overall score rather than by question or subset of questions. The full assessment without question 4 removed is included in the remainder of these results but we call to attention this potential limitation in interpreting the full results.

Demographic and Programming Experience Survey

Following the MCS1 and SCS1 assessments, students were asked to report both their demographic and programming experience information. For the demographic information, students were asked to report their age, gender, race/ethnicity, and primary language spoken in their childhood home. For the prior programming experience, students were asked to provide the following information. These questions were asked because they were a part of the standard SCS1 assessment that had been shared with the authors. Not all of the questions are used in this study and in subsequent offerings of this assessment we may modify the questions asked based on the results in this paper.

- FYE course (honors vs. standard)
- Which semester they took their FYE course

- Whether they took MCS1/SCS1 before or after their programming/computer science learning experience
- A Likert-scale to rate if the FYE course is their first programming experience
- A Likert-scale to rate their programming skills
- Their previous programming/computer science experiences
- Which programming languages they consider themselves minimally proficient in
- Whether they have used an online programming tutorials or resources (such as Code.org, Khan Academy, etc.)
- A Likert-scale to rate whether they would like to take more computing courses
- A Likert-scale to rate if they believe the skills taught in their FYE course will be useful in their life and/or career
- A Likert-scale to rate if they know how to use programming to communicate with others and/or other programmers
- How many times they have seen the MCS1/SCS1 assessment before

Results and Analysis

Prior High School Programming Experience

This subsection focuses on student responses to the following question: *What has been your previous programming or computer science experience(s)? (CS course in high school, CS course in college, workshop or professional development session, programming utility tools, Java Script for web design, Java Script for projects other than web design, self-taught, other)*. For this question, there were 1022 prior programming experiences reported by the 672 participants. The student responses are shown in Table 1 and Fig. 1.

Table 1: Prior Programming Experience Responses by Type

Type of Experience	Count	%
A computer science course at a high school	277	27.10
A computer science course at a college	136	13.31
A workshop or professional development session	23	2.25
Programming utility tools (Excel, calculators, etc.)	270	26.42
Java Script for web design	34	3.33
Java Script for projects other than web design	30	2.94
Self-taught via available resources	142	13.89
Other (please specify)	110	10.76

Since the most common prior programming experience was that students had taken a computer science course at a high school, these data were also broken out based on whether or not a student

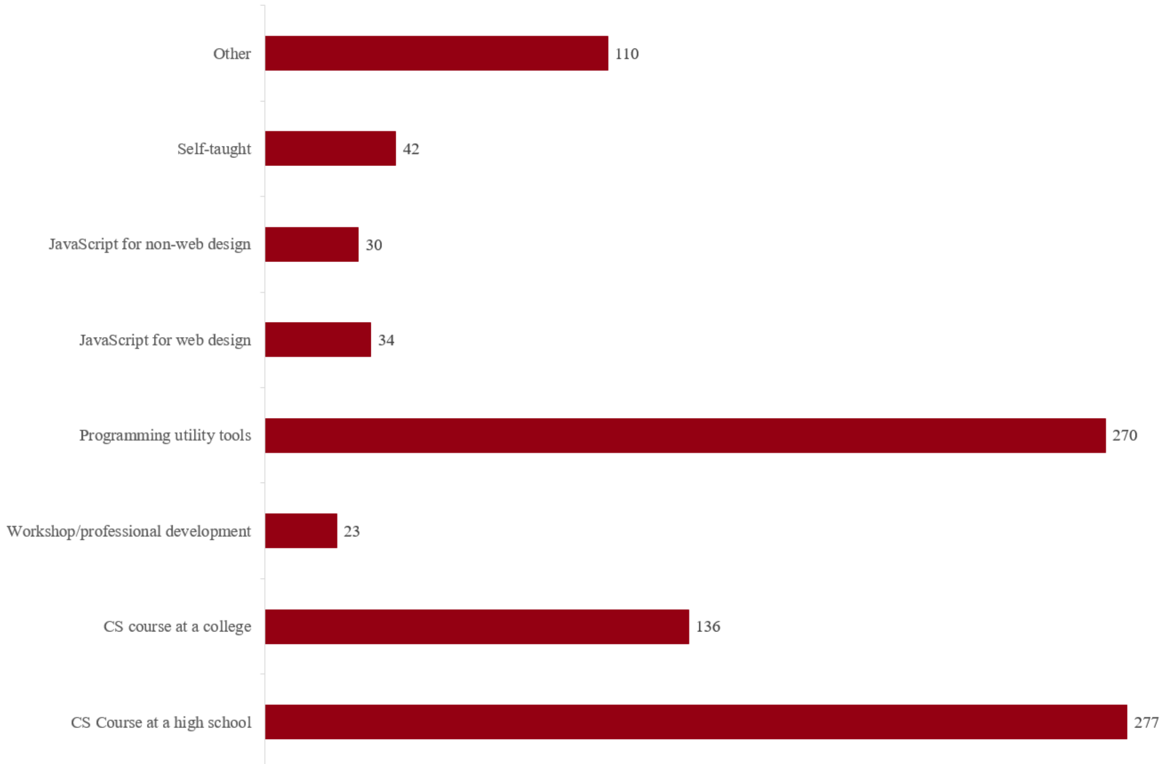

Figure 1: Prior Programming Experience by Type

reported taking a CS course in high school. There was no statistically different mean score on the assessments between students who took a CS course in high school and those who did not. As a result, while taking a CS course in high school was the most common prior experience reported by participants, it does not appear that, on average, these students scored any higher on the assessments.

An independent samples t-test was used to determine if there was a significant difference between the number of men and women who took CS in high school. A value of 1 represents that the student reported taking a CS course while a value of 0 corresponds to students who did not report taking a CS course in high school. With equal variances not assumed, the mean difference between men and women was 0.115 with a 2-tailed significance of 0.003. The difference in percentage of men and women who took a CS course in high school is shown in Fig. 2.

An independent samples t-test was also used to evaluate the mean difference in self-reported programming skill based on whether or not the student took a CS course in high school. For this question, students identified their skill level in a Likert scale with 1 corresponding to no programming skill and 5 corresponding to very strong programming skills. Assuming equal variances, a mean difference of 0.571 with a 2-tailed significance of 0.000 was found between students who had taken a CS course in HS and those who had not. Thus, students who took a CS course in high school typically reported that they had stronger programming skills. This result is also illustrated in Fig. 3. In the below chart, the red bars represent students who reported taking a CS course in high school while the grey bars represent students who did not report taking a high

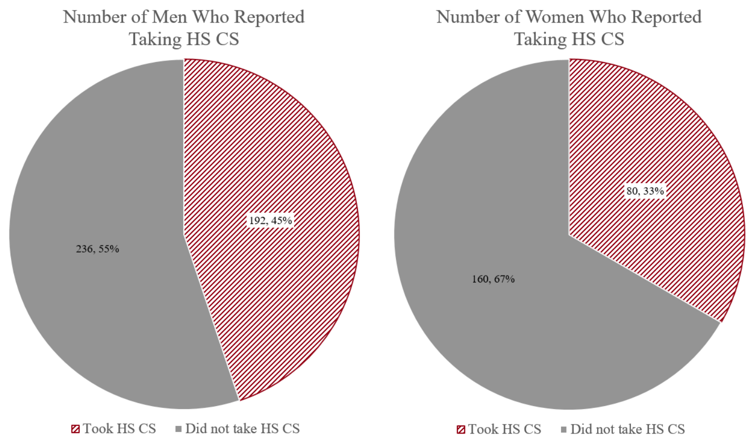

Figure 2: High School Computer Science Course Experience by Gender

school CS course. The distribution of the red bars is farther to the right (corresponding to a higher average reported skill level) than the gray bars indicating that students who took computer science in high school reported having better programming skills.

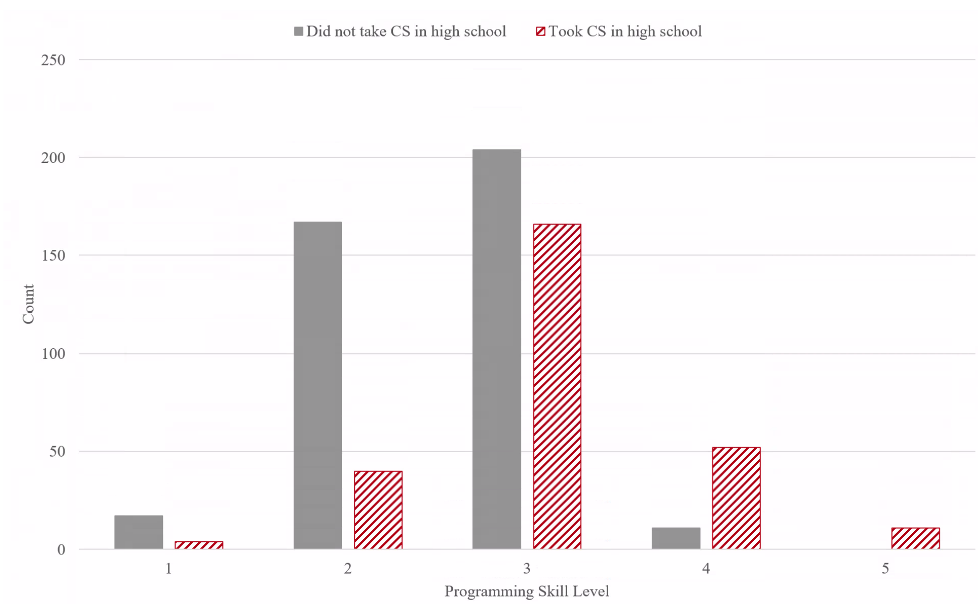

Figure 3: Self-Reported Programming Skill by High School Computer Science Experience

For the following analysis, the race/ethnicities were split into two categories: underrepresented minorities (URMs) and non-underrepresented minorities (non-URMs). Following the National Science Foundation's classification of underrepresented minorities in science and engineering, the American Indian or Alaskan Native, Black or African American, and Hispanic American groups are considered URMs for this study [14]. Asian/Pacific Islander and White/Caucasian are considered non-URMs. Using an independent samples t-test, the difference in prior programming experience by race was also evaluated. No significant difference was found between URM and non-URM participants regarding their prior programming experiences or whether they had taken a CS course in high school.

These results demonstrate that there are differences in prior programming experience in the first-year engineering students in these courses. Additionally, the differences in prior programming experience exist across genders. However, despite these differences in experience, there were not differences in the results in this concept inventory assessment demonstrating that students without prior programming experience at the end of their first semester course in engineering perform as well as their colleagues with prior experience. While this assessment was only given at the end of the semester, future assessments could be given at the beginning to see if the prior programming experience results in differences at the beginning of the course.

Self-reported Programming Skill

Related to prior programming experience, participants were also asked to rank themselves based on their programming skill. Since this assessment was at the end of the semester in a course where students learned programming, it is interesting to see that so many students self-report no or very little programming skills. Participants had the following options for identifying their skills.

1. I have no programming skills
2. I have very little programming skills
3. I have some programming skills
4. I have strong programming skills
5. I have very strong programming skills

The number of students who identified themselves with each category is shown in Table 2.

Table 2: Self-Identified Programming Skill of Participants

Skill Level	Count	%
No Skill	190	28.27
Very Little Skill	127	18.90
Some Skill	49	7.29
Strong Skills	107	15.92
Very Strong Skills	199	29.61

Using an independent samples t-test with equal variances not assumed, a statistically significant difference of 0.274 with 2-tailed significance of 0.000 was found between the mean reported skill of men and women. On average, men identified themselves as having strong programming skills while women identified themselves as having weaker programming skills. The descriptive statistics for average skill by gender is give in Table 3 and illustrated in Fig. 4.

This difference is expected given the extensive research that men are typically more confident in their math and science skills than women [15, 16, 17].

There were no statistically significant differences found in the self-reported skill between honors and standard students or between students who spoke English as a primary language in their

Table 3: Self-Reported Skill Statistics by Gender

Gender	N	Mean	Std. Deviation
Male	428	2.85	0.762
Female	240	2.58	0.635

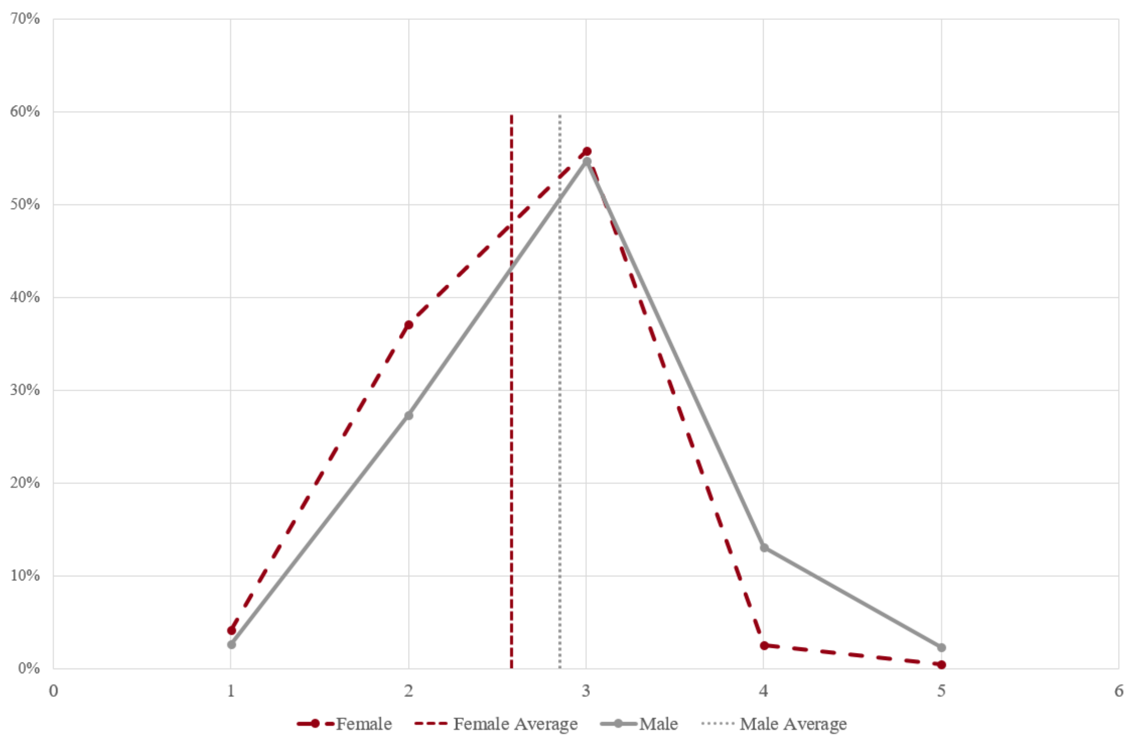

Figure 4: Self-Reported Skill by Gender

childhood home and students who spoke non-English languages. Additionally, there was no statistical difference in assessment scores between students who reported that they have strong or very strong programming skills and those who did not. This is an important finding that shows that self efficacy may not correlate to performance with these programming skills being tested.

Gender

The 672 responses were also divided by gender to determine how student scores differed by gender. Four participants did not select a binary gender. Given the small number of participants (<1%), this population was not investigated further. Table 4 contains the distribution of scores for males and females for MCS1 and SCS1 combined.

Table 4: MCS1 and SCS1 Combined Distribution of Scores for Females and Males

Gender	Sample Size	Mean Score	Std. Deviation
Male	428	38.00%	18.3%
Female	240	38.00%	16.4%

Table 4 shows that the mean score for males and females was effectively identical. An independent samples t-test confirmed that there was no statistical difference between the mean score of males and females.

Figure 5 shows the assessment scores for males and females.

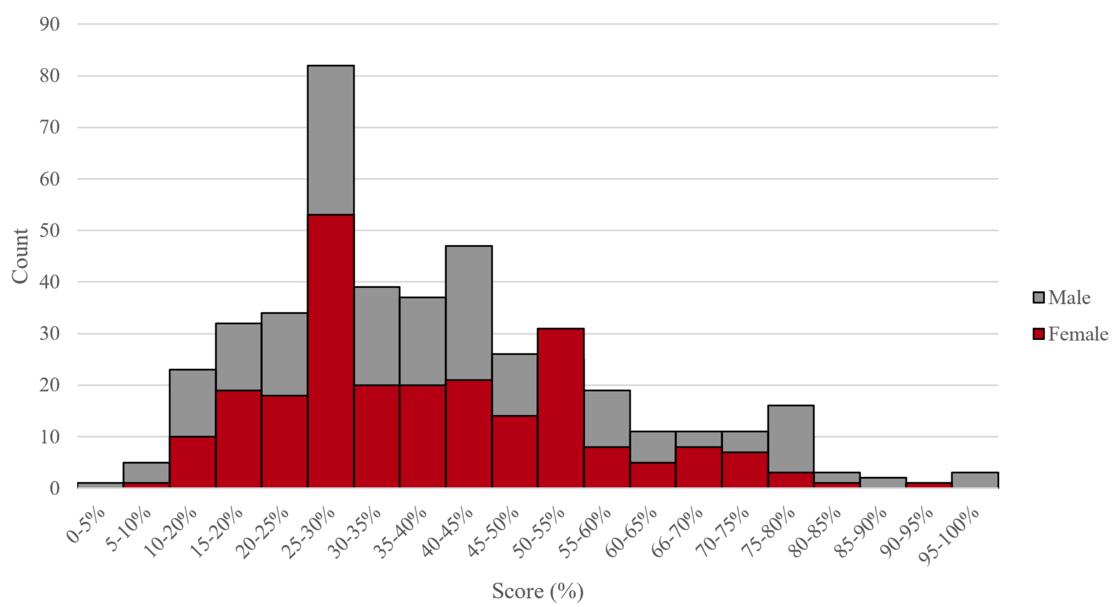

Figure 5: Frequency of scores for Male and Female Participants

Since there were no differences in assessment score by gender in the overall group, different markers were used to separate participants and determine if a gender difference existed within each group. An independent samples t-test was to assess these differences. The results of these t-tests are shown in Table 5. None of the results within any group had a significant difference in score between female and male students. This demonstrates that the assessment does not exhibit bias with respect to gender and that students in these courses have similar results at the end of the semester regardless of gender or prior experience.

Table 5: Gender Differences Within Groups

	Male		Female		
Group	**n**	**Mean(%)**	**n**	**Mean(%)**	**p-value**
Standard	329	36.05	185	35.44	.201
Honors (Combined)	99	46.37	55	46.24	.055
Honors	82	41.98	49	44.02	.255
Honors Advanced	17	67.55	6	64.35	.590
Strong Programming Skill	66	42.50	7	40.87	.215
Other Programming Skill	361	37.71	233	37.82	.777
High School Experience	192	39.54	80	38.51	.626
No High School Experience	236	37.54	160	37.61	.403
Underrepresented Minority	41	42.88	30	36.16	.129
Non-Underrepresented Minority	387	37.97	210	38.16	.560
SCS1	211	32.62	123	32.98	.940
MCS1	217	44.09	117	43.09	.157

Course Track: Honors or Standard

All participants were enrolled in one of two different courses, a standard and an honors course. Table 6 contains the distribution of scores combining MCS1 and SCS1.

Table 6: MCS1 and SCS1 Combined Distribution of Scores for Standard and Honors

Course Type	Sample Size	Mean Score	Std. Deviation
Standard	517	35.92%	16.70%
Honors	155	46.34%	18.53%

The score data for each course is illustrated in Fig. 6. The graph displays the percentage of each group that scored within 5% bins. Based on the shape of each distribution, it appears that the Standard course students tended to score slightly lower than the Honors course students on both assessments.

Using an independent samples t-test with equal variances not assumed, the appearance of the distribution was confirmed with a mean difference of 10.43% and a 2-tailed significance level of 0.000. This aligns with the expectation that students in a course that covers more programming languages and content perform better on a coding assessment than students in a course that covers less content.

Figure 6: Frequency of scores for Standard and Honors Course

The Honors course is split into two separate categories: regular or advanced. To enroll in the advanced sections of the Honors course students are required to have had prior programming experience before taking the course in order to handle an increased workload through extra extension programming assignments. The descriptive statistics for the regular Honors participants and the advanced Honor participants are shown in Table 7.

Table 7: MCS1 and SCS1 Combined Distribution of Scores for Regular Honors and Advanced Honors

Course Type	Sample Size	Mean Score	Std. Deviation
Honors (Regular)	132	42.80%	16.21%
Honors (Advanced)	23	66.72%	18.18%

By examining the means, it is expected that advanced Honors students performed better on MCS1 and SCS1. The distribution of scores on MCS1 and SCS1 for regular Honors and advanced Honors participants shown in Fig. 7 show a similar implication. By comparing the percent of each group, it is clear that the advanced Honors participants seem to perform better on these assessments than regular Honors participants.

By conducting an independent samples t-test, the mean difference between the regular Honors and advanced Honors participants' scores, with equal variances not assumed, was 23.92% with a 2-tailed significance of 0.000. This mean difference is substantial and is most likely due to the increased workload of additional programming assignments and practice.

Due to the large mean difference between regular and advanced Honors students, the analysis of

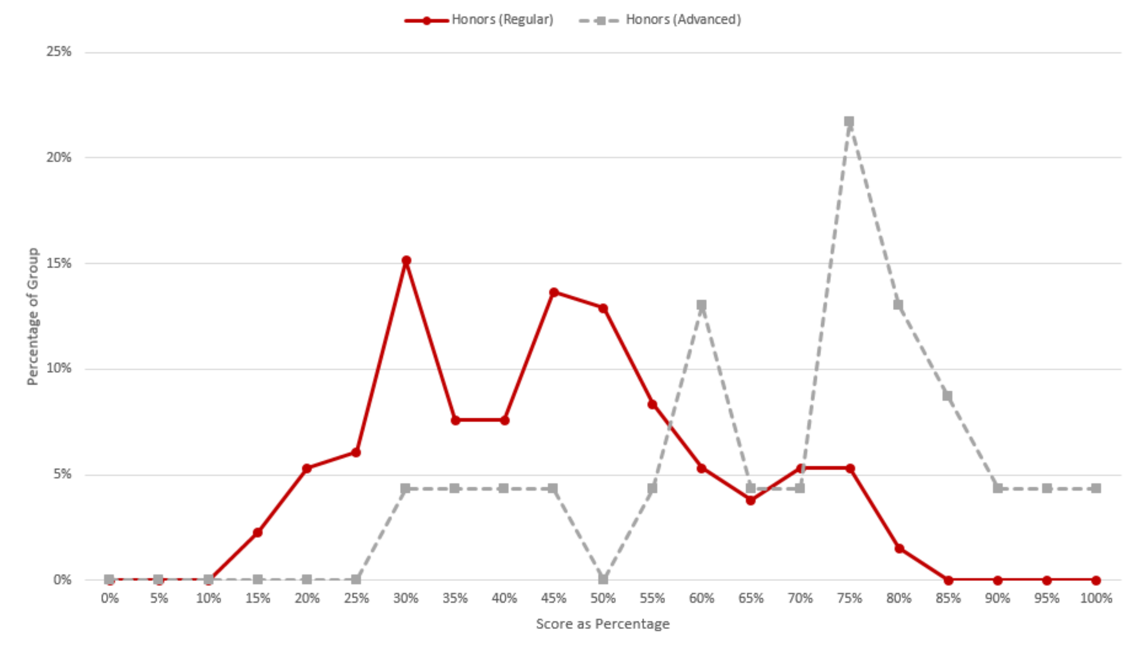

Figure 7: Frequency of scores for Honors (Regular) and Honors (Advanced) Course

the Honors and Standard course mean score differences must be reexamined without the advanced Honors students' scores impacting the distribution of scores. The distribution of the scores of Honors participants remains mostly unchanged with the exclusion of the advanced Honors participants because of the relatively small sample size. The score distribution between regular Honors students and Standard students is shown below in Figure 8.

After the exclusion of the advanced Honors students, the mean difference between Honors and Standard participants shrank to 6.88% with a 2-tailed significance of 0.000 from 10.43%. The change in mean difference is relatively small despite the large mean difference in scores between regular and advanced Honors participants. This is most likely due to the small sample size of advanced Honors course participants.

These results demonstrate that the course tracks do impact the results on the assessment and indicate that those in the honors track are ending the semester with higher scores on these conceptual assessments. This is potentially due to the increased workload and practice that is given in the honors course. However, since students are meant to have a similar knowledge of programming fundamentals after this course, these differences could be a concern in future engineering courses that require this knowledge. Having this common assessment is an important tool moving forward since currently the two courses have different assignments and exams.

Conclusion and Future Work

Two concept inventories, MCS1 and SCS1, were given to 672 first-year engineering students at Ohio State at the end of the autumn 2019 semester. Results have shown that students who had prior programming experience in high school ranked their programming skills higher than those who did not, but they did not perform statistically better on the concept inventory assessment. The

Figure 8: Frequency of scores for Standard and Honors (Regular) Course

same is true for males and females. Males rated their programming skills higher than females and also indicated higher levels of programming experience, however there was no statistically significant difference in their performance on the concept inventory assessment. While prior high school experience did not change performance on the assessment, there was a statistically significant difference between those in the honors course and those in the standard course. Those in the honors course performed about 10% higher on both assessments. Additionally, those in the advanced honors section performed higher on the assessment by about 24% than those in the regular honors section. The students in the honors course received essentially an entire extra programming course (an additional 3 credit hours) compared to those enrolled in the standard course and the advanced section received extra programming extension assignments. This highlights that the additional instruction could be a significant factor in the increases seen.

These concept inventory assessments can be used in the future as a pre- and post-test for the course to examine learning gains through the semester. These original results can also be used as a benchmark as curricula changes occur. Many of the engineering disciplines at this university have subsequent courses that include programming experiences but they are sometimes years after this original introduction. These assessments could be used in a longitudinal study to determine the knowledge loss that occurs after not practicing programming for a period of time.

References

[1] J. I. Smith and K. Tanner, "The problem of revealing how students think: Concept inventories and beyond," *CBE - Life Sciences Education*, vol. 9, 2017.

[2] ——, "The problem of revealing how students think: Concept inventories and beyond," *CBE - Life Sciences Education*, vol. 9, 2017.

[3] D. Hestenes, M. Wells, and G. Swachamer, "Force concept inventory," *The Physics Teacher*, no. 30, 1992.

[4] J. Libarkin, "Concept inventories in higher education science," *STEM Education Workshop 2*, pp. 1–10, 2008.

[5] J. C. Libarkin, S. W. Anderson, and B. Callen, "Development of the geoscience concept inventory," *Proceedings of the National STEM Assessment Conference*, pp. 148–158, 2006.

[6] R. S. Lindell, E. Peak, and T. M. Foster, "Are they all created equal? a comparison of different concept inventory development methodologies," in *AIP Conference Proceedings*, vol. 883, 2007.

[7] A. Yadav, D. Burkhart, E. Snow, P. Bandaru, and L. Clayborn, "Sowing the seeds of assessment literacy in secondary computer science education: A landscape study," 07 2015.

[8] A. E. Tew and M. Guzdial, "The fcs1: A language independent assessment of cs1 knowledge," in *SIGCSE '11 Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*. ACM, 2011.

[9] R. Caceffo, S. Wolfman, K. S. Booth, and R. Azevedo, "Developing a computer science concept inventory for introductory programming," in *SIGCSE Proceedings 2016*, 2016.

[10] A. E. Tew, "Assessing fundamental introductory computing knowledge in a language independent manner," Ph.D. dissertation, Georgia Institute of Technology, 2010.

[11] M. C. Parker, M. Guzdial, and S. Engleman, "Replication, validation, and use of a language independence cs1 knowledge assessment," in *ICER '16 Proceedings of the 2016 ACM Conference on International Computing Education*. ACM, 2016, pp. 93–101.

[12] A. E. Barach, C. Jenkins, S. S. Gunawardena, and K. M. Kecskemety, "MCS1: A matlab programming concept inventory for assessing first-year engineering courses," in *2020 ASEE Virtual Annual Conference Content Access*. Virtual On line: ASEE Conferences, June 2020, https://peer.asee.org/34958.

[13] "2019 college of engineering annual statistical report," The Ohio State University, Tech. Rep., 2019.

[14] National Science Foundation, National Center for Science and Engineering Statistics, "Women, minorities, and persons with disabilities in science and engineering: 2019," 2019, special report NSF 19-304.

[15] P. Ring, L. Neyse, T. David-Barett, and U. Schmidt, "Gender differences in performance predictions: Evidence from the cognitive reflection test," *Frontiers in Psychology*, vol. 7, 11 2016.

[16] L. G. Jones and L. P. Jones, "Context, confidence and the able girl1," *Educational Research*, vol. 31, no. 3, pp. 189–194, 1989. [Online]. Available: https://doi.org/10.1080/0013188890310304

[17] L. S. Dix, Ed., *Women: Their Underrepresentation and Career Differentials in Science and Engineering: Proceedings of a Workshop*. National Academy Press, 1987.