MCS1: A MATLAB Programming Concept Inventory for Assessing First-Year Engineering Courses

Ada E Barach, The 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 will be pursuing a Ph.D. in Computer Science in the fall.

Connor Jenkins, The 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, The 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.

Dr. Krista M Kecskemety, The 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.

MCS1: A MATLAB Programming Concept Inventory for Assessing First-Year Engineering Courses

Abstract

This complete research paper discusses the development of a new MATLAB-specific concept inventory, called MCS1, for assessing foundational computer science concepts as well as the preliminary data analysis from a pilot test of MCS1. Concept inventories are typically multiple-choice assessments for evaluating student understanding of specific concepts and are used across various STEM fields. Few computer science concept inventories have been developed due to a unique set of challenges such as computer science's heavy reliance on syntax and notation. Though other computer science concept inventories exist, they are largely language-independent which has been shown to favor high-performing students. As a result, the First-Year Engineering (FYE) program at The Ohio State University does not have many tools to assess student comprehension of MATLAB programming skills, teaching methods, or course curricula. MCS1 is a replication of a previously validated language-independent concept inventory for foundational computer science, called SCS1, developed by Parker et. al. Think-aloud interviews were conducted to determine if revisions were necessary before giving the assessment to current FYE students. Preliminary validation has focused on comparing the new MCS1 to the existing SCS1 through a point biserial correlation test but has found a statistically significant difference between MCS1 and SCS1 scores. This indicates that MCS1 cannot be validated against SCS1 and an independent validation study for MCS1 is necessary.

Introduction

A concept inventory is "an outline of core knowledge and concepts for a given field and a collection of multiple-choice questions that are designed to probe student understanding of these fundamental concepts" [1]. The first and most famous concept inventory, called the Force Concept Inventory (FCI), was developed as a diagnostic test for force concepts in physics and was published by Hestenes et. al in 1992 [2]. As the use of concept inventories grew in popularity due to the success of FCI [3, 4], more assessments were developed in areas such as physics, chemistry, astronomy, geoscience, and others [5]. Concept inventories are often given before and after instruction (referred to as the 'pre-test' and 'post-test' in the literature) [4]. This demonstrates the use of concept inventories as effective assessment tools. Faculty can gain insight into student understanding and develop teaching and assessment techniques [6].

In 2011, Allison Tew and Mark Guzdial developed the Foundational Computer Science 1 (FCS1) concept inventory for evaluating student understanding of basic programming knowledge [7]. FCS1 was a landmark concept inventory in computer science since so few had been developed [8]. Then, in 2016, Parker et. al replicated FCS1 to create the Second Computer Science 1 (SCS1) assessment. Parker et. al argued that creating more concept inventories allows instructors to better

assess their students' understanding and reduces the saturation (the availability of the questions and answers) of existing assessment tools. Following in Parker et. al's footsteps, MCS1 is a isomorphic copy of SCS1 using the same replication process [9].

In this study, a new MATLAB concept inventory assessment tool in foundational computer science for use in first-year engineering programs was developed. Concept inventories are common in many fields, especially the sciences [10, 11, 12], but computer programming has relatively few assessments in general [13]. The advantage of a concept inventory is that it can be a standard, valid assessment tool that is able to capture conceptual understanding. One of the challenges in creating a concept inventory for computer programming is that there are a variety of programming languages used and the nature of programming can make it difficult to determine if one is assessing the concept or the syntax of the language [14]. These challenges are some of the reasons that language-independent concept inventories for computer programming have been developed [7, 8, 9].

The First-Year Engineering (FYE) program at The Ohio State University lacks a validated assessment tool to determine student understanding of MATLAB programming concepts for first-year students. It is critical for FYE programs to have these tools available to allow the program and its instructors to determine the impacts of various curricular changes. The current programming language independent concept inventories have limitations for the type of computer programming taught to first-year engineering students at Ohio State. MATLAB, which is what is taught at this university, is a unique programming language with features not present in other languages such as Java and Python. These features make it difficult to test student knowledge with a language independent assessment. An example of one of these unique features in MATLAB programming is array indexes start at 1 compared to traditional programming languages that start at 0.

The goal of this research is to develop and validate a A MATLAB-specific concept inventory, MCS1, by replicating a previously validated foundational language-independent computer science concept inventory, SCS1 [7, 8, 9]. This paper presents the initial phase of this work, the development of the new assessment questions, think-aloud interviews, and preliminary data from piloting the assessment.

MCS1 has the potential to impact thousands of students enrolled in FYE courses annually by normalizing the assessment process for students. Further, this assessment can be incorporated into the curriculum and used by faculty and administrators to make informed decisions about the curriculum and programming instruction.

Methods

FYE Course Context

The context chosen for this work is the first-year engineering program at The Ohio State University. The first-year engineering program is a two-semester course sequence with the focus of this study being on the first-semester course. The first semester focuses on problem solving using computational tools, specifically programming. There are two distinct tracks of the course, an honors and a standard track, that have different number of students, contact time, credit hours,

and content. These differences are shown in Table 1. Both tracks begin teaching programming fundamentals through MATLAB. The honors course then teaches C/C++ programming following the MATLAB instruction.

Table 1: Course Information

Course	Students per Section	Credit Hours	Contact Time per Week (min)	Num Students AU 19	Programming Languages
Honors	36	5	375	408	MATLAB and C/C++
Standard	72	2	110	1480	MATLAB

Assessment Development

MCS1 was developed by creating an isomorphic copy of each SCS1 question. To do this, each question and its response options were converted from pseudocode to MATLAB. This isomorphic copy keeps the question style and content unchanged for MCS1 to ensure the concepts were tested in the same manner as SCS1. However, different variable names, function names, and values were used. SCS1 questions that did not have any written code in the question or the answers were not changed for MCS1 because the question could not be altered without affecting the question style or concept. An example of the translation from SCS1 to MCS1 questions is shown in Figure 1. Three questions from SCS1 which test recursion were not included in MCS1 since recursion is not part of the curriculum of the FYE course being considered for this study.

The 24 questions which make up MCS1 each test one of the following topics: arrays, basics, for loops, logical operators, function parameters, function return values, if statements, or while loops. Since MCS1 has 24 questions and eight topics, there are three questions that test each topic. Of these three questions, one is a definitional question type, one is tracing, and the other is code completion.

Think-Aloud Interviews

A think-aloud interview is a research method used to examine participants' thought process and logic while performing high-level cognition [15]. This technique was used to gather information from a limited number of participants while they took a completed draft of MCS1. During the interviews, participants had 90 minutes to answer as many questions as they could while talking through their problem-solving process for each question.

Six think-aloud interviews were conducted with a single participant present at each. The participants were chosen based on the order of response to an invitation sent out to all students in the FYE programs and then by their availability. The students were offered a gift card as a participation incentive. Of the six participants, three were students in the standard FYE course while the other three were students in the honors FYE course. Each student started at a different point in the test to ensure that each question was answered by at least one student in each course sequence.

Given the function definition:

```
DEFINE add(a,b,c)

    max = 0
    IF ((a < b)AND(a < c) THEN
        max = b + c
        RETURN max
    ELSE IF ((b < a) AND (b<c) THEN
        max = a + c
        RETURN max
    ELSE
        max = a + b
        RETURN max
    ENDIF
    PRINTLN max
ENDDEF
```

Which of the following statements best describes the execution of the function call

```
add(8,5,12);
```

A. Value of max = 13
B. After printing the max product, `add` will return the max
C. Nothing will ever be returned from this function.
D. Nothing will ever be printed from this function.
E. An error will be generated for attempting to print after returning max

Given the function definition:

```
function [diff] = subtract(a, b, c)
    if (a > b) & (b > c)
        diff = a - c;
    elseif (b > a) & (a > c)
        diff = b - a;
    else
        diff = c - a;
    end
end
```

Which of the following statements best describes the execution of the function call

```
subtract(5, 3, 7);
```

A. Value of `diff = 4`
B. After calculating `diff`, `subtract` will print `diff`
C. Nothing will ever be returned from this function.
D. Nothing will ever be printed from this function.
E. An error will be generated since `diff` is not returned.

Figure 1: Example mapping from SCS1 (left) to MCS1 (right)

Two researchers were present at each interview so one researcher could take notes while the other could prompt the participant for further thought and explanation if needed. The interviews were audio-recorded for the duration of the testing session and then reviewed to make note of anything missed during the interview. The notes and recordings were reviewed by a researcher not present at the initial session to help eliminate any internal bias. These notes were then compiled to resolve any typos, formatting issues, or revisions that needed to be made to MCS1.

The think-aloud interview data was scored using a modified version of Tew's rubric [8]. In the development of FCS1, Tew used this rubric to evaluate student responses in the think-aloud interviews for FCS1. While MCS1 is a replication of SCS1 and not FCS1, SCS1 was created to test the same concepts and use the same wording as FCS1. As a result, FCS1 and MCS1 should test the same concepts in a similar manner. Though a direct comparison between FCS1 and MCS1 cannot be made, the content analysis completed for FCS1 provides a benchmark for what one would expect to see in the think-aloud interviews for MCS1. The modified rubric is shown in Table 2.

Table 2: Rubric used to score MCS1 think-aloud interview responses [8]

Response Code*	Description
1	Participants answered question correctly by reasoning about intended construct
2	Participant answered question incorrectly by following common misconception or using faulty logic about construct
3	Participant answered question correctly even though they had incorrect reasoning about construct
4	Participant answered question correctly, however the correct answer was reached by reasoning about other conceptual content
5	Participant answered question incorrectly due to reasoning about other constructs
6	Participant answered question incorrectly. The wording of the question led to confusion/incorrect answer
8	Participant answered question incorrectly. The reasoning was incoherent and difficult to assign to any particular concept/construct

* Response code 7 is omitted as it refers to the transfer of knowledge to pseudocode which is not applicable to MCS1.

Response code 7 discusses transfer of knowledge from a specific language to pseudocode and was excluded from the rubric as it is not applicable to a single programming language assessment like MCS1. Each participant's data was reviewed independently by two different researchers. The scores were then compared and discussed until a consensus was reached about which response code was most appropriate.

Assessment Piloting and Analysis

To compare MCS1 and SCS1, data for both tests was needed. All FYE students were contacted with the opportunity to take part in the study and were incentivized with extra credit in their FYE course if they participated. In total, there were 672 usable participant responses from 21 hour-long testing sessions.

During the testing sessions, participants were automatically and randomly given MCS1 or SCS1 by the testing software once given access by the proctor. The participants were limited to 60 minutes to complete the assessment at which time their testing session would end and they would be redirected to the demographic portion of the survey. Those who were given SCS1 were also given the pseudocode guide that was included in the initial testing of SCS1 [8]. At the completion of each testing session, the survey closed to prevent any submissions outside of the proctored assessment times. The participants also received their scores broken down by category soon after submitting the tests.

Results and Analysis

Think-Aloud Interview Results

Once the think-aloud interviews were completed, two different analysis methods were used to determine how students interpreted the MCS1 questions and where revisions needed to be made. Using the audio recording and researcher notes from each interview, a list of potential revisions was created based on the type of revision reported by participants. Figure 2 below shows a portion of this list.

#	Revision Category	Issue	# Participants Reporting	Resolved?	Notes
1	Question Revision	Removed "at the beginning" from choice A	2	Yes	
2					
3					
4	Question Revision	Update to have E be the correct answer	4	Yes	
4	Question Revision	Remove "of 4 characters" in code comment.	Researcher	Yes	
4	Typo	Make "not" in question bold and italicized to match other questions	Researcher	Yes	
5	Question Revision	Inconsistency between double quotes in question vs. single quotes in the function definition	1	Yes	
5	Typo	Removed parentheses in if-loop to be more consistent with other questions	Researcher	Yes	
5	Formatting	Change "concatentation" in question to be default font.	Researcher	Yes	
6	Question Revision	Conditional operators is defined in SCS1 pseudo code but not in MCS1	3	Yes	Add description in question

Categories
Answer Revision
Question Revision
Formatting
Typo

Figure 2: Revisions made from think-aloud interview data

As shown in the above figure, revisions were categorized into one of four types:

1. Answer Revision - The response options for a question were incorrect or confusing to students.
2. Question Revision - The question stem was poorly worded or incorrect.
3. Formatting - The formatting of the question stem and/or response options was inconsistent with the rest of the assessment.
4. Typo - The question stem or response options contained a typographical error.

In some cases, the researchers conducting the interview noticed errors that either did not affect or were not reported by participants. Next, a content analysis was conducted to categorize participants' responses to each question. Table 3 details the results of this analysis.

The results from the content analysis were then compared to the content analysis completed during the creation of FCS1 [8] and is shown in Figure 3. For this purpose, the responses of interest were categories 1 and 2 as these show that the responses in the think-aloud interviews

Table 3: Rubric used to score MCS1 think-aloud interview responses.

Response Code	Description	%
1	Participants answered question correctly by reasoning about intended construct	62.71
2	Participant answered question incorrectly by following common misconception or using faulty logic about construct	27.12
3	Participant answered question correctly even though they had incorrect reasoning about construct	3.39
4	Participant answered question correctly, however the correct answer was reached by reasoning about other conceptual content	5.08
5	Participant answered question incorrectly due to reasoning about other constructs	2.54
6	Participant answered question incorrectly. The wording of the question led to confusion/incorrect answer	0.85
8	Participant answered question incorrectly. The reasoning was incoherent and difficult to assign to any particular concept/construct	0.00

were connected to the construct that each question was testing. As seen in Figure 2 , MCS1 had a higher incidence of these responses at 89.8% versus FCS1 with 83% desirable responses. MCS1 also had 5.1% of responses in category 4 as opposed to FCS1 which had 0%. It is hypothesized that this difference could be due to differing interpretations of the response codes between the MCS1 and FCS1 researchers.

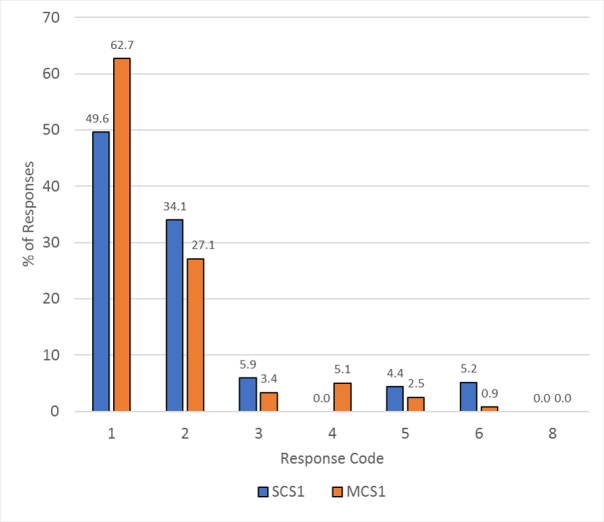


Figure 3: Think-aloud response code comparisons.

Preliminary Analysis of Assessment Pilot

The first step to analyzing the data was removing incomplete responses. Responses were not removed if they skipped questions, but they were removed if the participants did not complete or attempt sections of the survey, including the demographic questions. This resulted in 679 responses. To account for participants who rushed to complete the assessment rather than seriously consider each question, the average time (2183.7s) and the standard deviation of time (812.3s) to complete the survey were used to create criteria for removing participants. Participants who completed the test over two standard deviations faster than the average (less than 559s) were removed. This resulted in 672 responses, consisting of 336 SCS1 participants and 336 MCS1 participants.

The descriptive statistics for the assessment testing is shown in Table 4. The average score for MCS1 was 43.89%, with a standard deviation of 19.36%. The average score for SCS1 was 32.78%, with standard deviation of 13.74%. While completing the preliminary analysis of data, it was found that Question 4 of SCS1 included a typo which meant that there were multiple correct answers to the question. As a result many students were recorded as incorrect on this question for not choosing the supposed correct answer. After removing Question 4, the average score for SCS1 was 33.61%, with standard deviation of 14.34%. This difference of 1% is not expected to change any of the comparisons to MSC1 since the difference between means of SCS1 and MCS1 is 10%. Subsequent data presented shows the full SCS1 with the supposed correct answers, but attention is called to this typo as needed to fully interpret the results. The distribution of scores for MCS1 and SCS1 can be seen in Figure 4.

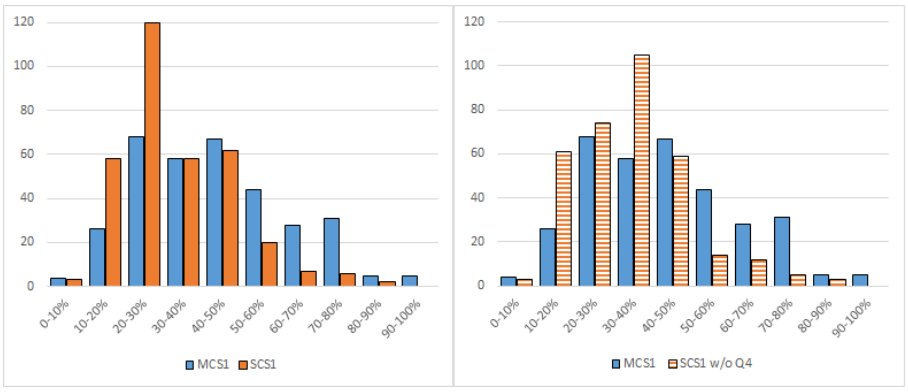
Table 4: Assessment testing statistics.

Test	Average Score (raw)	Average Score (%)	Standard Deviation (raw)	Standard Deviation (%)
SCS1 (n=336)	8.85 (out of 27)	32.78%	3.71	13.74%
SCS1 w/out Q4 (n=336)	8.74 (out of 26)	33.61%	3.73	14.34%
MCS1 (n=336)	10.53 (out of 24)	43.89%	4.65	19.36%

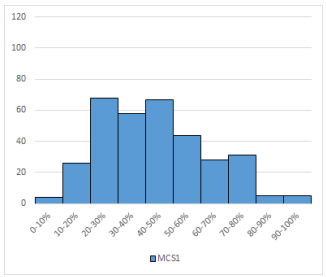
The distribution of scores for SCS1 is moderately skewed in the positive direction (skewness=0.908) whereas the distribution of scores for MCS1 is fairly symmetrical (skewness=0.415). With the removal of Question 4, SCS1 is still moderately skewed in the positive direction (skewness=0.937). The kurtosis for the SCS1 scores indicates a heavy-tailed distribution (kurtosis=1.022) and MCS1 scores have a light-tailed distribution (kurtosis= -0.435). After removing Question 4 the distribution remain largely unchanged (kurtosis=1.048).

The MCS1 data does not represent a normal distribution as required by many of the statistical tests. However, we believe it is reasonable to assume a normal distribution of MCS1 due to the skew, kurtosis values, and large sample size.

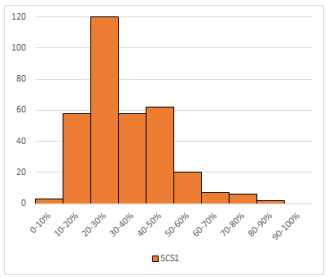
The goal of this study was to demonstrate that MCS1 produced similar results to SCS1 and therefore we could assume validity of the new assessment. Due to the method of testing, each



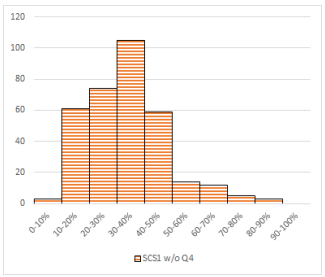
(a) SCS1 and MCS1



(b) MCS1



(c) SCS1



(d) SCS1 without Q4

Figure 4: Score distributions for both SCS1 and MCS1.

participant only completed one test (either MCS1 or SCS1). A point-biserial correlation was performed to find the strength of the association between score and the test students took (either MCS1 or SCS1) because the point-biserial correlation analyzes the relationship between a dichotomous variable and a continuous variable, in this case which test was taken and the score on the test [16]. There was a correlation between score and test, which was statistically significant ($r_{pb} = .314, n = 672, p < .001$). After removing Question 4, there was still a statistically significant correlation ($r_{pb} = .289, n = 672, p < .001$). These correlations are considered a statistically medium effect according to Jacob Cohen's standards for effect size and cannot be used to determine validity for MCS1 independently [17].

To compare the distributions, a Mann-Whitney U test, a non-parametric test for independent samples, was computed and a statistically significant difference was found between the SCS1 scores and the MCS1 scores ($p < .001$). This did not change with the removal of Question 4. Because their distributions are significantly different from one another along with lack of evidence to the contrary, MCS1 cannot be validated directly against SCS1 using this data [16].

Reliability

A Chronbach's Alpha test for reliability of the two assessments was completed. Additionally, the test was run on all questions to show which could be removed to improve the internal reliability of the assessment. For SCS1, the Chronbach's Alpha was 0.635. Table 5 shows the questions that, if removed, would increase the internal reliability of the assessment and to what extent the

Chronbach's Alpha value would increase by removing those questions.

Table 5: Effect of Removing Questions on Chronbach's Alpha Value

Assessment	Question	New Chronbach's Alpha Value	Change in Chronbach's Alpha Value
SCS1 $\alpha = 0.635$	4	0.641	0.006
	5	0.637	0.002
	18	0.640	0.005
	20*	0.640	0.005
	27†	0.779	0.144
MCS1 $\alpha = 0.779$	18*	0.787	0.008
	24†	0.782	0.003

* Equivalent Question
† Equivalent Question

Question 20 on SCS1 is the equivalent question to Question 18 on MCS1 and Question 27 on SCS1 is equivalent to Question 24 on MCS1, so it is not surprising that they appear on both lists. Question 18 on SCS1 is one of the questions that is removed in MCS1. Therefore, the questionable items are similar between both tests with the exception of Question 4 and Question 5 in both assessments which only resulted in consistency issues for SCS1. Given the typographical error in Question 4 of SCS1, the consistency increase which results from removing the question is expected. For both of these tests, if individual items were removed the overall consistency only increases slightly. The overall consistency value for SCS1 of 0.635 is considered useful for group averages and 0.779 for MCS1 is considered fairly good for this type of test and good for group measurements[18].

Conclusion and Future Work

MCS1, a replication of SCS1, is among the first MATLAB-specific concept inventories for foundational computer science. Preliminary data indicates a statistically significant difference between SCS1 and MCS1 indicating that MCS1 cannot be validated against SCS1. The lack of a correlation between MCS1 and SCS1 also calls for a future study to provide a standalone validation of MCS1. Additionally, the typo present in SCS1 for this study also supports the standalone validation of MCS1.

Unlike SCS1, MCS1 is language-specific which may reduce score bias towards high-performing students [7]. With a Cronbach's Alpha of 0.779, which is considered acceptable [19], MCS1 improves upon the internal consistency of questions presented in SCS1 which had a Cronbach's Alpha of 0.635. Additionally, since the participants took either SCS1 or MCS1, MCS1 did not experience the learning effect found in Parker et. al's testing of SCS1[9].

Future work for this study entails a deeper analysis of the pilot data and specifically an independent validation study will be conducted to determine if MCS1 is valid on its own. This full validation study will explore if a correlation exists between it and student scores on the final exam for the FYE course. Lastly, the team will further examine the effect of honors vs. standard course sequence on MCS1 scores, the effect of prior programming experience, and if any gender differences exist.

Further replications, improvements, and development of language-specific concept inventories is encouraged. A wider variety of validated assessments that are available to instructors will provide potentially more accurate and context-appropriate tools to measure the effectiveness of classroom teaching methods. For example, a Python-specific assessment may be more useful than MCS1 to an FYE program that teaches Python as its introductory programming language. MCS1 could also be expanded upon to test more concept areas or to test more MATLAB-specific features such as graphing or matrix operations.

Currently, MCS1 provides the groundwork for a new concept inventory for foundational computer science topics in the MATLAB programming language. The validity of an assessment is strongly affected by how easily the questions and answers can be found so to protect the potential validity of MCS1, the questions are not included here. Future studies will aim to fully validate MCS1 so that it can provide instructors at Ohio State with a tool to assess student comprehension in the FYE program. If validated, the MCS1 assessment will be made available to other instructors and institutions. It is hoped that MCS1 will be used to guide teaching methods at this university. MCS1 will potentially impact thousands of first-year students by providing instructors a method by which to assess and improve teaching strategies.

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

[3] D. Evans, G. L. Gray, S. Krause, J. Martin, C. Midkiff, B. M. Notaros, M. Pavelich, D. Rancour, T. Reed-Rhoads, P. Steif, R. Streveler, and K. Wage, "Progress on concept inventory assessment tools," in *33rd ASEE/IEEE Frontiers in Education Conference*, 2003, pp. T4G–1 – T4G–8.

[4] H. J. G. Sands, Parker, "Using concept inventories to measure understanding," 2018.

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

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

[7] 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.

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

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

[15] E. Charters, "The use of think-aloud methods in qualitative research: An introduction to think-aloud methods," *Brock Education*, no. 12, pp. 68–82, 2003.

[16] A. Field, *Discovering Statistics Using IBM SPSS Statistics*, 4th ed., M. Carmichael, Ed. Sage Publications Ltd, 2013.

[17] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. Routledge, 1988.

[18] R. Doran, *Basic Measurement and Evaluation of Science Instruction*. National Science Teachers Association, 1980.

[19] D. George and P. Mallery, *IBM SPSS Statistics Step by Step: A Simple Guide and Reference*, 14th ed. Routledge, 2016.