Validation of the Placement Skill Inventory: A CS0/CS1 Placement Exam

Ryan Bockmon
ryan.bockmon@unl.edu
University of Nebraska–Lincoln
Lincoln, NE, USA

Chris Bourke
chris.bourke@unl.edu
University of Nebraska–Lincoln
Lincoln, NE, USA

ABSTRACT

Student success in introductory computing course continues to be a major challenge. Though there has been much research and innovation in recent years to help reduce high failure rates, a substantial population of students still struggle in a typical CS1 course. In this paper we create an argument of validity of the Placement Skills Inventory (PSIv1). The goal of the PSIv1 is to help advise and place students into an appropriate introductory computing course. While placement exams have been developed in the past, the goal of PSIv1 is to differentiate students who will be successful in a CS1 course and those that would be better served taking a CS0 course as their first computing course. In contrast, traditional placement exams have focused on differentiating students between CS1 and CS2. The PSIv1 is a combination of two instruments, the Computational Thinking Concepts and Skills Test and the Second Computer Science 1 Exam Revised Version 2. These two instruments measure students' computation thinking skills and prior programming knowledge respectively. The PSIv1 was administered to all students enrolled in either a CS0 or CS1 during the first two weeks of the semester. We use Item Response Theory to create an argument of validity of the PSIv1 and look at differences in scores on the PSIv1 based on if a student passed or failed a CS0 and CS1 course. We then used the results to create an advising strategy and criteria to help students decided if they should enroll in a CS0 or CS1 course.

CCS CONCEPTS

• **Social and Professional topics** → **Student assessment**.

KEYWORDS

CS0, CS1, Assessment, Validation, IRT

ACM Reference Format:
Ryan Bockmon and Chris Bourke. 2023. Validation of the Placement Skill Inventory: A CS0/CS1 Placement Exam. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2023), March 15–18, 2023, Toronto, ON, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3545945.3569738

SIGCSE '23, March 15–18, 2023, Toronto, ON, Canada.

ACM ISBN 978-1-4503-9431-4/23/03…$15.00
https://doi.org/10.1145/3545945.3569738

1 INTRODUCTION

Student success in introductory computing courses continues to be a major challenge [20]. While a majority of students have positive outcomes, there is still a substantial population of students who struggle in a typical CS1 course. There has been much effort and success in exposing adolescent and young adults to programming either by taking a course in high school or participating in outreach programs. However, there is still a significant number of incoming freshman that have never been exposed to programming. In our experience this has caused a rift with our students. Though we believe that all students, regardless of prior programming/computing experience, should have as likely of a chance to succeed in CS1, we have observed that students without some prior experience are less confident and less motivated to succeed compared to students that have had prior experience. It is our view that a significant number of these students would be more successful if instead they had the opportunity to take a more introductory course such as CS0 as their first college-level computing course.

The typical instrument for enrolling students into an appropriate introductory level course in many disciplines is a placement exam. Within computing disciplines placement exams [4, 11, 12, 14, 18] have been developed but have focused on distinguishing between students with significant prior programming experience (who may be enrolled in a CS2 course) and those with little or no experience (who are thus enrolled in CS1). These exams have been designed to gauge a student's knowledge in topics that are typically covered in a CS1 course and so represent a post-hoc measure of their learning outcomes rather than an instrument that can be used to *predict* student success in a CS1 course.

Other tests have attempted to gauge student ability in algorithmic and computational thinking [10, 21]. While these skills are essential to success in a computing program, they do not necessarily provide a useful instrument to determine which introductory computing course a student should take.

Student Attitude Surveys [9, 15, 16] are another instrument that have been developed. This is especially important because student attitudes play a key role in success in learning computing [23], in particular student self-efficacy [22].

1.1 Research Background

In 2018 the School of Computing at the University of Nebraska-Lincoln consolidated its CS1 courses into a single large course serving Computer Science and Computer Engineering majors as well as non-majors from other disciplines [8]. The course was revamped to emphasize collaboration and active programming exercises and delivered in a hybrid manner (with parallel in-person and online sections). Over the last several years fall sections have had a consistent enrollment of over 200. Though the delivery of the course has been successful, student outcomes (in particular DFW rates) have not improved over the typical academia-wide averages [20].

The School of Computing had offered a CS0 course (using Python) which served as a service course for non-majors as well as a computing minor/major recruiting course. However, the curriculum did not grant credit for this course for majors and so CS1 was the de facto introductory course for computing majors.

Two initiatives were made to more fully utilize our CS0 course to improve overall student outcomes. First, curricular changes were made so that CS0 would count toward all computing degree programs including our minor. Specifically, students who took CS0 as their first computing course would be granted credit for it as a technical elective upon successful completion of other requirements. This change ensured that taking CS0 would not be a waste of credit hours or delay graduation.

Second, the School of Computing set about adopting, adapting and developing a placement exam intended to distinguish between students who would likely be successful in our CS1 course and those who would be better served (and more successful longterm) by taking our CS0 course as their first computing course instead. The goal of this study is to report on the development of this CS0/CS1 placement exam and to provide a rigorous validation of it.

2 RELATED WORK

Over the past several years there has been a great focus on creating and developing CS instruments and surveys that measure students' programming and computational thinking (CT) skills. This section is split into two parts. Part one covers several CS1 content instruments and part two covers several CT instruments.

2.1 CS1 Content Tests

One of the first CS content instruments is the Advanced Placement Computer Science (AP CS) exam [4]. The exam was first created in 1984 and was meant to test the equivalent of a first-semester course in computer science. The current version of the AP CS exam is written for Java and included multiple choice and free-response questions. Because the AP CS exam was written in Java, there was limited use of the exam. Tew et al.'s Foundational CS1 Assessment instrument (FCS1) was an attempted to create a CS assessment test independent of any single programming language [19]. The FCS1 tests student ability using pseudocode written in the style of imperative languages [9, 19].

A successor of the FCS1, the Second CS1 (SCS1) exam by Parker et al. [12] was created out of concern that the test would be saturated or that the answers could easily be found online if too many people used the test. In 2019, Bockmon et al. revised the SCS1 (called the SCS1R [6]) down to 9 questions to try to reduce the estimated completion time to around 30 minutes while also retaining the reliability of the SCS1.

In 2020 Peteranetz et al. worked on validating the Nebraska Assessment of Computing Knowledge (NACK) [14]. The goal of the NACK was to have scores be roughly normally distributed around 50% to measure students ability across an even distribution. The initial assessment consisted of 26 multiple choice questions and after three pilot testings, the NACK was refined down to 13 multiple-choice questions.

2.2 Computational Thinking Tests

In 2012 Werner et al. created the Fairy performance assessment to measure computational thinking in middle school students [21]. The Fairy assessment was created in Alice and was meant to assess two out of the three areas of computational thinking defined by the Carnegie Mellon Center for Computational Thinking: algorithmic thinking and making effective use of abstract thinking. In 2014 Koh et al. worked on creating a Real Time Evaluation and Assessment of Computational Thinking (REACT) system [10]. REACT was created as an assessment system for the Salable Game Design project to enable teachers to see which high-level computational thinking concepts students had mastered and which ones they were struggling with in real time.

Gonzalez worked on creating the Computational Thinking Test in 2015 [17]. The Computational Thinking Test was aimed at 7th and 8th grade Spanish students. Zhong et al. created the Three-Dimensional Integrated Assessment for Computational Thinking (TDIA) [24]. The TDIA was created to integrate three dimensions (directionality, openness, and process) into the design of several assessment tasks and to comprehensively assess the three dimensions of computational thinking (concepts, practices, and perspectives).

Again in 2020 Peteranetz et al. worked on validating the Computational Thinking Concepts and Skills Test (CTCAST) [13]. This test differed from the NACK as it was created to measure components of CT rather than CS1 content. After revisions they conducted another study in which they ran both the NACK and the CTCAST at the same time. Results indicated that the CTCAST and the NACK measure similar, but not identical, aspects of students' knowledge and skills, and that item-level statistics vary according to the scoring method that is used.

3 METHODS

3.1 Data Collection

Data was collected during the fall 2021 semester at a large R1 university in the midwest United States. Surveys were administered during the first two weeks of the semester. To avoid participation bias [5] participation was mandatory for all students enrolled in either a CS1 or CS0 course. Students received some portion of their course grade based on completion of the surveys as either a lab, homework, or participation grade depending on instructor preference. Each section was taught in a different programming language: Java, C, Python or Matlab. There was a total of 334 students who completed the surveys across all CS1 courses and a total of 123 students who completed the survey in the CS0 course. Final letter grades for each student were then obtained after the semester ended. Everything was approved by the institution's IRB. Analysis of final letter grades were calculated separately between the CS1 students and the CS0 students to compare differences between them.

3.2 Placement Skills Inventory Design

Two different instruments were used for this study. The first was the Computational Thinking Concepts and Skills Test (CTCAST) created and validated by Peteranetz et al. [13]. The CTCAST consisted of 12 multiple choice questions with 4 to 5 possible answers for each question. The second was the Second Computer Science 1 Exam Revised version 2 (SCS1Rv2). The SCS1Rv2 was created because during the validation process of the SCS1R it was found to be a difficult test and it was hard to distinguished students' programming aptitudes based on the instrument itself [6]. Both instruments were selected as they have been individually validated on a subset of students at the university this study took part.

This newer version removed one question from the SCS1R that was determined to have a bad fit and added four questions (thought to be easier) from the AP Computer Science exam [4]. The goal was to better determine students' programming skills across a larger range. These four questions were converted to pseudocode to keep consistency with the questions from the SCS1R. The SCS1Rv2 consists of a total of 12 multiple choice questions with 5 possible answers for each question. Both instruments were administered together as a single instrument as a CS0/CS1 placement exam. We call this combined instrument the Placement Skills Inventory version 1 (PSIv1). The PSIv1 consist of 24 multiple choice questions. The first 12 questions of the PSIv1 are from the CTCAST and the Second 12 questions are from the SCS1Rv2. The estimated time for completion of the PSIv1 is around one hour to ninety minutes.

4 ANALYSIS

Both a t-test and a non-parametric Mann-Whitney U test were used to test for differences in scores between two groups: students who received a failing grade (a final letter grade of a C- or below) or passed (a final letter grade of C or above) in their course. Both the t-test and a Mann Whitney-U test were used for redundancy and consistency across all data sets.

Item Response Theory (IRT) was used to validate the PSIv1 on both the CS0 and CS1 data sets combined. IRT is used to understand each question's difficulty and fit. Difficulty of a question is determined by how many students correctly answer that question and the fit of a question is determined by how well that question is at distinguishing between low ability students and high ability students. A good test should have multiple questions at different scales of difficulties. This allows the examiner to measure an examinee's performance across a large range of skills. For item difficulty we used conditional maximum likelihood (CML) estimation [2, 3]. CML scores range from -3 to 3. A CML closer to -3 means that the item is relatively easier, 0 means average difficulty, and 3 is the hardest difficulty.

To visualize the range of item difficulties we used an Item Characteristic Curve (ICC). The ICC represents the probability of an examinee obtaining the correct response to a question. An item with a difficulty of -0.5 means that a examinee with an ability of -0.5 will have a 50% chance of getting that question correct. The Likelihood Test Statistic and z-value were used to provide a measurement of each item's fit. A significant z-value indicates the item difficulty parameters differ across the raw score groups and that the IRT model does not hold for that question [3]. Desirable values are routinely accepted to be those where $|z| \leq 2$ [7]. For a more in depth tutorial with equations on how IRT is used on a computing instrument can be found in [6].

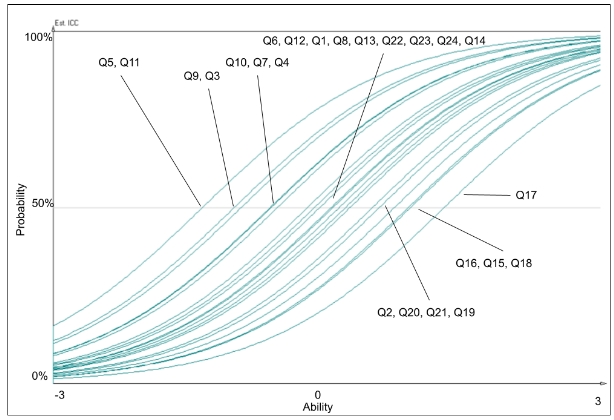

Figure 1: Item Characteristic Curve (ICC) plot for each of the 24 questions of the PSIv1

5 RESULTS

5.1 IRT Results

Table 1 displays the difficulty and fit for each question in the PSIv1. The "Total" column grouping reports data for all students. This data is broken down into two subsets in the column groupings, "Low Abilities" (students who scored below relative average) and "High Abilities" (students who scored above relative average). Within each grouping, we report the number of students who got the question correct (n+), the total percent of students who got the question correct (% correct), and the conditional maximum likelihood/item difficulty of the question (CML). The final column reports the z-value (fit) of each question.

Results in the "Total" column grouping indicate a large range of question difficulties. The most difficult question was Q17 with a CML of 1.28 and a total correct response rate of only 20.2% (essentially guessing). The easiest questions were Q5 and Q11 with a CML of a -1.364 and a total correct response rate of 74.5%. The ICC plot (Figure 1) visualizes the range of difficulties for each question in order of least-to-most difficult (left-to-right). Q5, Q11, Q9, Q3, Q10, Q7, and Q4 were relatively easy questions with a CML ranging in [-1, -0.5]. Q6, Q12, Q1, Q8, Q13, Q22, Q23, Q24, and Q14 were all relatively moderate questions with a CML around 0. Q2, Q20, Q21, Q19, Q16, Q15, Q18, and Q17 were relatively harder questions with a CML ranging in [0.5, 1.2].

Results from the z-value column indicate that several questions have a poor fit ($|z| \geq 2$). Q1, Q2, Q5, Q6, Q9, Q17 and Q18 all had absolute z-values greater than 2 and should be either revised or removed in future iterations of this instrument.

5.2 Grade Distribution

5.2.1 CS1 Grades. Figure 2 displays box plots of students' raw scores on the PSIv1 grouped based on their final letter grade (C and above or below a C) in a CS1 course. The left most graph shows the total combined score of the PSIv1 for a total possible score of a 24. Students who failed the course had a mean score of a 8.97, median score of a 9.0, and a standard deviation of a 3.71. Students who passed the course had a mean score of a 11.59, median score

Table 1: Item Response Theory Results of the PSIv1.

	Total (n = 455)			Low Abilities (n = 260)			High Abilities (n = 195)			z-value
Question	n+	% correct	CML	n+	% correct	CML	n+	% correct	CML	
Q1	215	47.3	-0.079	102	39.2	-0.266	113	57.9	0.209	-2.400
Q2	157	34.5	0.499	77	29.6	0.163	80	41.0	0.895	-3.593
Q3	302	66.4	-0.940	145	55.8	-0.942	157	80.5	-0.885	-0.255
Q4	263	57.8	-0.543	125	48.1	-0.629	138	70.8	-0.354	-1.348
Q5	339	74.5	-1.364	156	60.0	-1.118	183	93.8	-2.186	3.358
Q6	229	50.3	-0.213	116	44.6	-0.489	113	57.9	0.209	-3.550
Q7	270	59.3	-0.612	128	49.2	-0.676	142	72.8	-0.454	-1.070
Q8	202	44.4	0.047	87	33.5	-0.015	115	59.0	0.167	-0.905
Q9	309	67.9	-1.016	137	52.7	-0.816	172	88.2	-1.476	2.601
Q10	271	59.6	-0.622	118	45.4	-0.520	153	78.5	-0.760	1.107
Q11	339	74.5	-1.364	165	63.5	-1.268	174	89.2	-1.578	1.181
Q12	221	48.6	-0.137	90	34.6	-0.067	131	67.2	-0.186	0.584
Q13	200	44.0	0.066	79	30.4	0.127	121	62.1	0.038	0.435
Q14	181	39.8	0.253	62	23.8	0.459	119	61.0	0.081	1.795
Q15	122	26.8	0.892	49	18.8	0.757	73	37.4	1.046	-1.318
Q16	135	29.7	0.740	51	19.6	0.708	84	43.1	0.810	-0.475
Q17	92	20.2	1.286	40	15.4	1.001	52	26.7	1.548	-2.308
Q18	120	26.4	0.916	53	20.4	0.660	67	34.4	1.181	-2.389
Q19	150	33.0	0.574	53	20.4	0.660	97	49.7	0.540	0.558
Q20	157	34.5	0.499	60	23.1	0.501	97	49.7	0.540	-0.186
Q21	150	33.0	0.574	48	18.5	0.782	102	52.3	0.437	1.584
Q22	193	42.4	0.135	71	27.3	0.277	122	62.6	0.016	1.259
Q23	186	40.9	0.204	70	26.9	0.296	116	59.5	0.145	0.732
Q24	186	40.9	0.204	64	24.6	0.417	122	62.6	0.016	1.908

of 12.0 with a standard deviation of a 3.98. Both the t-test (t score = 5.0, p < 0.01) and Mann-Whitney U test (U score = 12969, p < 0.01) showed significant differences of scores based on if a student passed or failed their CS1 course.

The middle graph displays box plots of students' raw scores (out of 12) on the CTCAST. Students who failed the course had a mean score of a 6.04, median score of 6.0, and a standard deviation of 2.61. Students who passed the course had a mean score of a 7.26, median score of 8.0, and a standard deviation of a 2.21. Both the t-test (t score = 3.95, p < 0.01) and Mann-Whitney U test (U score = 12386, p < 0.01) showed significant differences of scores based on if a student passed or failed a CS1 course.

The right graph displays box plots of students' raw scores (out of 12) on the SCS1Rv2. Students who failed the course had a mean score of a 2.93, median score of 3.0, and a standard deviation of a 1.86. Students who passed the course had a mean score of a 4.33, median score of 4.0, and a standard deviation of a 2.51. Both the T-test (t score = 4.42, p < 0.01) and Mann-Whitney U test (U score = 12483, p < 0.01) showed significant differences of scores based on if a student passed or failed a CS1 course.

5.2.2 CS0 Grades. Figure 3 displays box plots of students' raw scores on the PSIv1 grouped based on their final letter grade (C and above or below a C) in a CS0 course. The left most graph shows the total combined score of the PSIv1 for a total possible score of a 24. Students who failed the course had a mean score of 10.81, median score of 10.0, and a standard deviation of a 3.66. Students who passed the course had a mean score of 10.72, median score of 11.0, and a a standard deviation of 3.65. Both the t-test (t score = -0.11, p = 0.91) and Mann-Whitney U test (U score = 1122.5, p = 0.94) did not show significant differences of scores based on if a student passed or failed a CS0 course.

The middle graph displays box plots of students' raw scores (out of 12) on the CTCAST. Students who failed the course had a mean score of 7.64, median score of 8.0, and a standard deviation of a 2.27. Students who passed the course had a mean score of 7.21, median score of 8.0, and a standard deviation of 2.29. Both the t-test (t score = -0.77, p = 0.44) and Mann-Whitney U test (U score = 1012, p = 0.51) did not show any significant differences of scores based on if a student passed or failed a CS0 course.

The right graph displays box plots of students' raw scores (out of 12) on the SCS1Rv2. Students who failed the course had a mean score of 3.2, median score of 3.0, and a standard deviation of a 2.31. Students who passed the course had a mean score of 3.5, median score of 3.0, and a standard deviation of 2.15. Both the t-test (t score = 0.92, p = 0.53) and Mann-Whitney U test (U score = 1232.5, p = 0.42) did not show any significant differences of scores based on if a student passed or failed a CS0 course.

6 DISCUSSION

6.1 IRT

Overall there was a large range of question difficulties across the PSIv1 indicating that the PSIv1 can measure students abilities across a larger range. When splitting the PSIv1 into each original instrument, we see that the majority of the CTCAST questions (Q1-Q12) were considerably easier compared to the SCS1Rv2 questions (Q13-24). The CTCAST difficulties ranged from mid 30% to mid 70% while

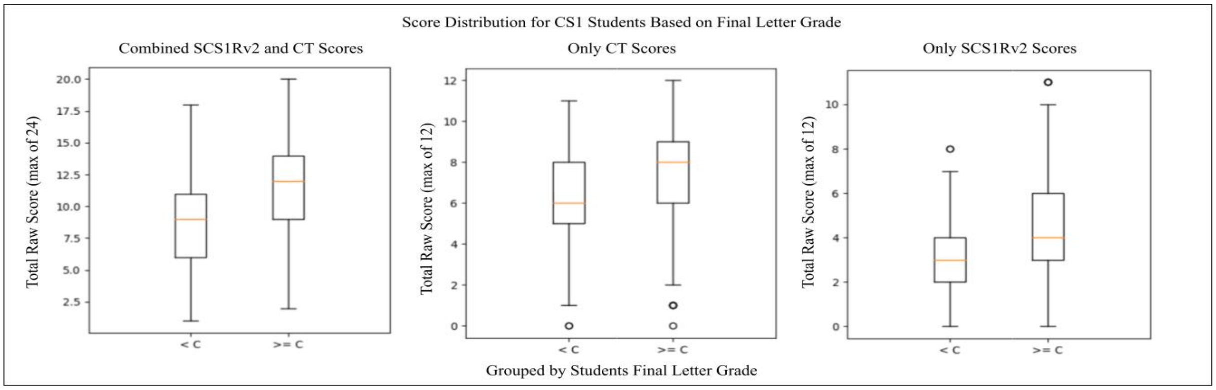

Figure 2: Box plots of students’ scores on the PSIv1 (left), on the CTCAST (middle), and SCS1Rv2 (right) grouped based on their final letter grade in a CS1 course

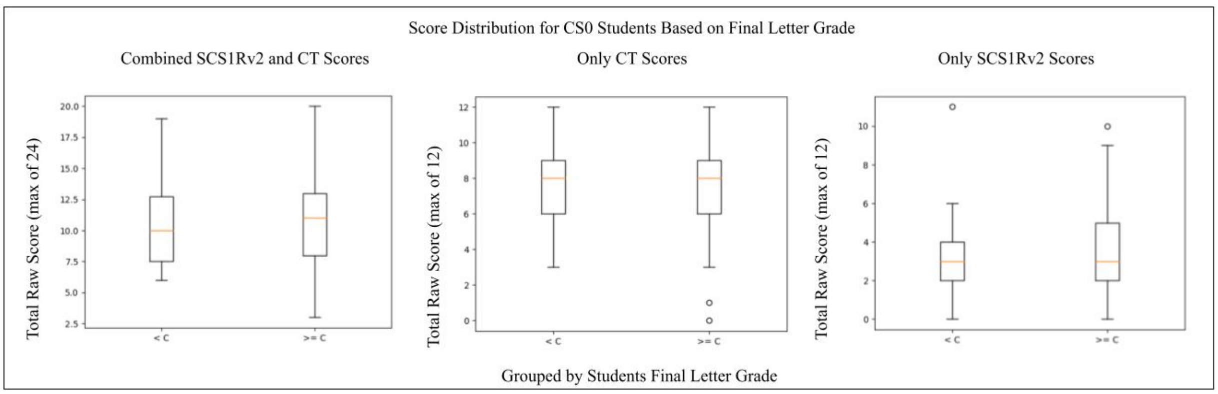

Figure 3: Box plots of students’ scores on the PSIv1 (left), on the CTCAST (middle), and SCS1Rv2 (right) grouped based on their final letter grade in a CS0 course

the SCS1Rv2 difficulties ranged from the low 20% to mid 40%. This makes sense because we are measuring students prior to them taking any college introductory computing course and did not expect students to have any programming experience when participating in this study.

One of our ancillary goals was to add several questions to help make the SCS1Rv2 slightly easier to increase the range of question difficulties in order to identify students at a larger range of skills compared to the SCS1R. Q21, Q22, Q23, and Q24 were all new questions that were added to the SCS1Rv2. Based on the results, we conclude that all were relatively easier than original questions (Q13-Q20). However, the SCS1R was validated using pre- and post-course scores while the SCS1Rv2 has only been administered as a pre-course exam. Further studies are currently being conducted to validate the SCS1Rv2 on both pre- and post-course scores.

Q1, Q2, Q5, Q6, Q9, Q17, and Q18 were all found to have a poor fit. Q17 and Q18 were considered the hardest questions with only a total of correct response rate around guessing (20%). We are not surprised that both these questions had poor fit either. In general, both low ability students and high ability students scored close to the same on both these questions. With low ability students having a correct response rate right at or below guessing and high ability students having a correct response rate slightly above guessing. Thus, it is difficult to distinguish student abilities based solely on these two questions. In future iterations of the PSIv1 these questions should either be revised or removed.

The one outlier is Q5. Low ability students had a total correct response rate at 60% and high ability students had 93.8%. A considerably large difference between the two. However, the issue lies with its CML value. Q5 CML for both low ability students and high ability students is relatively large and the larger the CML value is, the less likely it is that the model fits the data and thus reflected in the z-value [1].

6.2 Grade Distribution

We did not see any difference between students who passed or failed a CS0 course and how well they did on the PSIv1. We also did not see any difference between students who passed or failed a CS0 course when splitting the PSIv1 between the CSCAST and SCSRv2. This is what we had hoped for because that means it does not matter what prior knowledge students have coming into a CS0 course. Each student is as likely to pass the course based on their prior computational thinking and programming knowledge (if any).

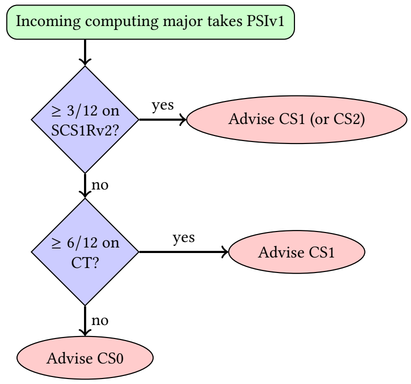

Figure 4: Advising based on placement exam results.

We did however see a significant difference between students who passed or failed a CS1 course and how well they did on the PSIv1. Students who passed the course scored higher on the PSIv1 compared to students who failed the course. When we investigated further we saw that on average students who failed the course scored around guessing (3 out of 12 correct) on the SCS1Rv2. In other words, they likely. did not have any prior programming experience. This makes sense as students who already have prior programming experience would probably have a higher chance at passing an introductory course. These students might also be able to even test out of a CS1 course and enroll in a CS2 course (the traditional motivation for administering a placement exam). We also observed that between students who passed or failed a CS1 course there was a significant difference on how well they scored on the CTCAST. Students who passed the course had an average higher score on the CTCAST compared to students who failed the course. These results indicate that there is some predictable capability of the PSIv1 and student success in a CS1 course.

6.3 Advising Strategy

Based on the results of this study, the School of Computing adopted an advising strategy in the summer of 2022 (for fall 2022 enrollment). The PSIv1 was administered to all incoming computing majors (taking it was mandatory for all new students prior to enrollment). However, it was not labeled as a "placement exam" but as the title (Skills Inventory) in order to reduce any anxiety a student felt about taking an "exam". Further, students were informed that the results would only be used to place them in the most appropriate introductory course to maximize their success (in order to increase motivation and so that they would take the exam seriously). Based on a student's results, advisors were directed to enroll the student in one of either CS0, CS1, or CS2 (see Figure 4).

If the student scored more than 3/12 on the SCS1Rv2, they were advised to take CS1 as their first computing course. If, in addition, the student indicated they had a strong prior programming background (during an advising session), the advisor would have a follow up discussion with the student to determine if CS2 was a better fit.[1]

If a student did not score more than 3/12 on the SCS1Rv2, their score on the CTCAST was considered. If they scored 6/12 or more, it indicated a stronger computational thinking ability and so were still advised into CS1 (but not considered for CS2). If a student did not meet either threshold, they were advised to enroll in a CS0 course as their first computing course.

We note that the advising process itself is mandatory (all incoming students must be advised prior to enrollment), it is of course *advisory*; students are still free to enroll in any course for which they meet an official prerequisite which means that a student is free to enroll in either CS0 or CS1.

It is hoped that the validation of our PSIv1 instrument can be repeated in fall 2022 and that the adopted advising strategy will lead to both short-term (individual course) and long-term (major matriculation) student success.

7 LIMITATIONS

This study has several limitations. First, as a pilot study, these results are based on only one university and may have different results at other institutions. Secondly, to avoid participation bias [5] we made participation mandatory for all students enrolled in either a CS0 or CS1 course. However, students were only graded based on if they completed the "exams" or not. Students had no incentive to try to complete the PSIv1 to the best of their ability other than their own motivation. It is possible that this exam only measures students' self-motivation. In this case we still believe that these results are useful to help determine if a student should enroll in a CS0 or a CS1 course. We also acknowledge that this exam is written only in English and could also be testing a persons ability to understand the English language. It may be that the student does not comprehend the question even if they would be able to answer it correctly if written in their primary language.

8 CONCLUSION & FUTURE WORK

The creation and validation of instruments is an ongoing process. The results of the IRT model indicate that several questions should be revised or removed and we plan to continue to improve the PSIv1 as we continue collecting data.

Research is already being conducted to investigate the extent that this exam, course, and other factors that have been shown to be a predictor of student success (such as self-reported programming skills, motivation, concerns, expected letter grade) impact student retention rates. With the full pilot in the School of Computing, in fall 2022, we hope to gather more data in the actual implementation of PSIv1 and the above advising strategy. We anticipate conducting a full multi-year study and intend to extend this instrument and its study to other institutions.

ACKNOWLEDGMENTS

We would like to thank the School of Computing Curriculum Committee and Dr. Suzette Person. Without their help we would not have been able to conduct this study.

[1]This typically involves giving the student a sample of the first CS2 assignment consisting of some basic programming exercises from CS1 and gauging the student's confidence in completing the exercises in the programming language(s) they are familiar with. Most students tend to self-select into CS1 after such a discussion.

REFERENCES

[1] [n.d.]. Conditional Likelihood. https://www.sciencedirect.com/topics/mathematics/conditional-likelihood. Accessed: Now 1, 2022.

[2] E. Andersen. 1973. A goodness of fit test for the Rasch model. *Psychometrika* 38, 1 (1973), 123–140.

[3] F.B. Baker and S.H. Kim. 2004. *Item Response Theory: Parameter Estimation Techniques, Second Edition*. Taylor & Francis.

[4] College Board. 2010. *2009 AP computer science a released exam*. The College Board, New York.

[5] R. Bockmon and S. Cooper. 2022. What's Your Placebo? The Dangers of Participation Bias. *Commun. ACM* Volume 65, 10 (2022), 31–33. https://doi.org/10.1145/3528085

[6] Ryan Bockmon, Stephen Cooper, Jonathan Gratch, and Mohsen Dorodchi. 2019. (Re)Validating Cognitive Introductory Computing Instruments. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) *(SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 552–557. https://doi.org/10.1145/3287324.3287372

[7] T.G. Bond and C.M. Fox. 2007. *Applying the Rasch Model: Fundamental Measurement in the Human Sciences*. Lawrence Erlbaum Associates Publishers.

[8] Chris Bourke. 2021. *Development & Evolution of a Computer Science I Course*. Technical Report. University of Nebraska-Lincoln. https://digitalcommons.unl.edu/prtunl/202/

[9] Allison Elliott Tew, Brian Dorn, and Oliver Schneider. 2012. Toward a Validated Computing Attitudes Survey. In *Proceedings of the Ninth Annual International Conference on International Computing Education Research* (Auckland, New Zealand) *(ICER '12)*. Association for Computing Machinery, New York, NY, USA, 135–142. https://doi.org/10.1145/2361276.2361303

[10] Kyu Han Koh, Ashok Basawapatna, Hilarie Nickerson, and Alexander Repenning. 2014. Real Time Assessment of Computational Thinking. *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC*, 49–52. https://doi.org/10.1109/VLHCC.2014.6883021

[11] L. Lacher, A. Jiang, Y. Zhang, and M. Lewis. 2017. Aptitude and previous experience in CS1 classes. *Int'l Conf. Frontiers in Education: CS and CE* 17 (2017), 87–91.

[12] Miranda C. Parker, Mark Guzdial, and Shelly Engleman. 2016. Replication, Validation, and Use of a Language Independent CS1 Knowledge Assessment. In *Proceedings of the 2016 ACM Conference on International Computing Education Research* (Melbourne, VIC, Australia) *(ICER '16)*. Association for Computing Machinery, New York, NY, USA, 93–101. https://doi.org/10.1145/2960310.2960316

[13] Markeya S Peteranetz and Anthony D Albano. 2020. Development and Evaluation of the Nebraska Assessment of Computing Knowledge. *Frontiers in Computer Science* 2 (2020), 11.

[14] Markeya S. Peteranetz, Patrick M. Morrow, and Leen-Kiat Soh. 2020. *Development and Validation of the Computational Thinking Concepts and Skills Test*. Association for Computing Machinery, New York, NY, USA, 926–932. https://doi.org/10.1145/3328778.3366813

[15] V. Ramalingam, D. LaBelle, and S. Wiedenbeck. 2004. Self-efficacy and mental models in learning to program. *ACM SIGCSE Bulletin* 36, 3 (2004), 171–175.

[16] V. Ramalingam and S. Wiedenbeck. 1998. Development and validation of scores on a computer programming self-efficacy scale and group analyses of novice programmer self-efficacy. *Journal of Educational Computing Research* 19, 4 (1998), 367–381.

[17] M. Román González. 2015. Computational Thinking Test: Design Guidelines and Content Validation. In *EDULEARN15 Proceedings* (Barcelona, Spain) *(7th International Conference on Education and New Learning Technologies)*. IATED, 2436–2444.

[18] Allison Elliott Tew and Mark Guzdial. 2010. Developing a Validated Assessment of Fundamental CS1 Concepts. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education* (Milwaukee, Wisconsin, USA) *(SIGCSE '10)*. Association for Computing Machinery, New York, NY, USA, 97–101. https://doi.org/10.1145/1734263.1734297

[19] Allison Elliott Tew and Mark Guzdial. 2011. The FCS1: A Language Independent Assessment of CS1 Knowledge. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education* (Dallas, TX, USA) *(SIGCSE '11)*. Association for Computing Machinery, New York, NY, USA, 111–116. https://doi.org/10.1145/1953163.1953200

[20] Christopher Watson and Frederick W.B. Li. 2014. Failure Rates in Introductory Programming Revisited. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (Uppsala, Sweden) *(ITiCSE '14)*. Association for Computing Machinery, New York, NY, USA, 39–44. https://doi.org/10.1145/2591708.2591749

[21] Linda Werner, Jill Denner, Shannon Campe, and Damon Chizuru Kawamoto. 2012. The Fairy Performance Assessment: Measuring Computational Thinking in Middle School *(SIGCSE '12)*. Association for Computing Machinery, New York, NY, USA, 215–220. https://doi.org/10.1145/2157136.2157200

[22] S. Wiedenbeck, B. LaBelle, and V. Kain. 2004. Factors affecting course outcomes in introductory programming. *Workshop of the Psychology of Programming Interest Group* (2004).

[23] Y. Zhang and S. Espinoza. 1988. Relationships among computer self-efficacy, attitudes toward computers, and desirability of learning computing skills. *Journal of Research on Computing in Education* 30, 4 (1988), 420–436.

[24] Baichang Zhong, Qiyun Wang, Jie Chen, and Yi Li. 2016. An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research* 53, 4 (2016), 562–590.