Can Students' Spatial Skills Predict Their Programming Abilities?

Ryan Bockmon
University of Nebraska - Lincoln
Lincoln, Nebraska
rbockmon@cse.unl.edu

Stephen Cooper
University of Nebraska - Lincoln
Lincoln, Nebraska
scooper22@unl.edu

Jonathan Gratch
Texas Woman's University
Denton, Texas
jgratch@twu.edu

Jian Zhang
Texas Woman's University
Denton, Texas
jzhang@twu.edu

ABSTRACT

Spatial abilities have been shown to have high predictability in students' success in STEM related fields. Studies have also shown that there is a correlation between students' spatial skills and programming abilities, but it is unknown how well students' prior spatial abilities can predict students' introductory programming abilities at the end of the semester. During this study we used a multinomal logistic regression to create a predictive model to predict students' introductory programming abilities at the end of the semester. The highest model accuracy (64.6%) was obtained when accounting for students' prior programming abilities, prior spatial skills, socioeconomic status, and three factors regarding students' attitudes towards computing. It was also found that when looking at the predictability of each individual variable, students' prior spatial ability had the highest predictability (56.6% accuracy) when compared to all other variables.

CCS CONCEPTS

• **Social and Professional topics → CS1**;

KEYWORDS

CS1; Intervention; Replication; Spatial Skills; Attitudes

ACM Reference Format:

Ryan Bockmon, Stephen Cooper, Jonathan Gratch, and Jian Zhang. 2020. Can Students' Spatial Skills Predict Their Programming Abilities?. In *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '20), June 15–19, 2020, Trondheim, Norway.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3341525.3387380

1 INTRODUCTION

Introductory computing courses are known to have high failure rates, [27] and the idea to build a model that can identify at-risk students is not new. Studies have looked at a multitude of reasons why students might fail. Clicker scores, online quizzes, gender, prior programming experiences, SAT scores, and achievement goals have all been used to build predictive models with relatively high accuracy [15, 29].

Recent studies have found that students' spatial skills have high correlations to students' programming abilities [3, 6]. It was even shown that with the use of a spatial skills intervention that was run alongside introductory computing courses, both students' spatial skills and programming abilities increased [2]. While others studies have shown that spatial skills are a good predictor of student success in STEM fields, little is known about how predictable students' introductory programming abilities are at the end of the semester when accounting for their prior spatial skills.

This paper attempts to explore the predictability of individual variables that were collected when gathering baseline data for a larger study intended to validate various assessments. These variables include: prior programming abilities, prior spatial skills, gender, race, socioeconomic stats (SES) and five factors regarding students' attitudes towards computing: problem solving - transfer, personal interest, problem solving - strategies, real world connections, problem solving - fixed mindset, which are further described in section 3.1 below. This paper also attempts to build a predictive model using a combination of each of those variables. We pose the following research questions to frame our work:

(1) How accurate is each individual variable at predicting students' programming abilities at the end of the semester?
(2) What variables contribute to a model with the highest accuracy?

2 RELATED RESEARCH

2.1 Factors Impacting CS1 Success

Several studies have examined what factors impact student success in computing courses [4, 14, 21, 29–31]. In 1986, Werth investigated the relationship between students' grades at the end of a CS1 course and their sex, age, high school performance, college performance, number of mathematics courses, work experiences, cognitive development, cognitive style and personality factors. Werth found that, overall, college grades, the number of hours worked, and the number of high school math classes taken had the most significant

ITiCSE '20, June 15–19, 2020, Trondheim, Norway

ACM ISBN 978-1-4503-6874-2/20/06...$15.00
https://doi.org/10.1145/3341525.3387380

relationship to course grades. Werth also found that how well students did on the group embedded figures test and the measure of Piagetian significantly correlated with grades in the course [29].

Wilson and Shrock ran a study during the spring semester of 2000 in a CS2 course. Their findings showed that students' comfort level and math grades had a positive influence on their success, while attribution to luck had a negative influence on their success [31]. Rountree et al. ran a study from 2000-2001 and collected survey responses from 472 students. Their results show that the only indicator of success from their surveys was whether or not a student was expecting to get an A from the course [21]. Byrne and Lyons ran a study with 110 students that showed a clear link between programming ability and existing aptitude in mathematics and science subjects [4]. Wiedenbeck et al. found that both students' mental models of programming and self-efficacy have a direct effect on their overall success in an introductory programming course [30]. Lacher et al. ran a study across six sections of CS1 in the fall of 2015 and found a correlation between students aptitude level and final course grade [14]. Bergin and Reilly ran a study from 2003-2004 and found that students', gender, comfort level, mathematics scores, science scores and perception of their understating of a computational module accounted for 79% of the variance in programming performance.

Most recently, in 2019 Liao et al. found that prerequisites and clicker responses provide high accuracy for predicting students' risk levels. They also found that assignments and online quizzes added some accuracy to these predictions [15]. When accounting for clicker questions, online quizzes and assignments their model was able to achieve up to 79% accuracy when predicting students' risk levels, where risk levels is the opposite of student success.

2.2 Spatial Skills and STEM

Spatial skills have been shown to be important in success in many engineering disciplines and in science [5, 23, 24]. Barker found that well developed spatial skills are essential for understand basic and structural chemistry [1]. Carter et al. ran a study with 2,498 students enrolled in the first semester of a college-level general chemistry course where they found that spatial score is most strongly correlated to unit conversion calculations [5]. Sorby found that a person's spatial skills have a correlation between their ability to learn to use computer aided design software [24]. Shea et al. ran a twenty year long longitudinal study with 563 thirteen year old children that scored at the top 0.5% on the Scholastic Assessment Test Mathematics and Scholastic Assessment Test Verbal, finding that the students' spatial ability added validity to the SAT-M and SAT-V in predicting educational outcomes over the 20 years [22].

2.3 Spatial Skills and Computing

In 1984, Webb ran a study with 35 students and found that students' spatial ability was the best predictor of knowledge of basic commands after learning Logo for one week [28]. In 1986, Mayer et al. ran a study with 57 college students in a course in Basic and found that logical reasoning and spatial abilities were the primary cause of success in learning Basic [16]. Vicent et al. found that a person's spatial skill level was the most significant predictor of success in their ability to interact with and take advantage of the computer interface in performing database manipulations [25]. Norman proposed models for how individual differences are expected to affect performance when technology is introduced stating that, "the primary cognitive factor driving differences in performance using computer-based technology is spatial visitation" [17].

Cox looked into a student's ability to navigate source code and the creation of their codespace or "mental model of source-code structure" [7]. Fisher et al. explored how sex differences linked to spatial cognition and codespace [11]. Fisher et al. also found that not only are there gender differences in spatial skills, but gender plays a role in how a person navigates source code. Males tend to use a top down development/comprehension strategy while females tend to use route-based bottom-up development/comprehension strategies [11].

Fincher ran a study with 177 participants from eleven post-secondary educational institutions and found a small positive correlation between scores in a spatial visualization task and programming marks, though attributing programming success to higher IQ rather than to spatial skills [10]. Jones and Burnett ran a study with 24 participants finding that participants with high spatial abilities completed code comprehension exercises faster than those with lower spatial abilities, along with a strong relation between spatial ability and results in their programming modules [12]. Jones and Burnett later found a correlation between mental rotation skills and programming success [13]. Most recently Parkinson et al. investigated the relationship between spatial skills and computer science [20] and Parker et al. found that students' spatial skills have a high impact on computer science achievement [19].

3 METHODS

In this study, we targeted students who were enrolled in introductory computer science courses at three universities in the USA: the University of Nebraska - Lincoln, Texas Woman's University and University of the North Carolina at Charlotte. Data was collected across multiple sections of CS1. Each section was taught in a different programming language: Java, C, Python or Matlab. We collected data by using a pre-post format during the fall semester of 2018 and spring semester of 2019. All participation was voluntary, with incentives used to encouraged students to participate. All incentives were approved by each university's IRB process and consisting of a $10 gift card and/or extra credit depending on what the instructor offered. We note that because participation in the study was voluntary that there is a chance of having participation bias in our results. There was a total of 197 participants, with 119 men, 72 women and 6 who identified as neither men nor women.

3.1 Instrument Design

Four instruments were used to collect data. The first instrument collected basic demographic data on a student: age, gender, race and socioeconomic status (SES). The second instruments used was the Revised Purdue Spatial Visualization Test (PSVT:R) [32]. The PSVT:R consists of 30 questions that present two 3-D objects depicted in 2-D, as illustrated in Figure 1. The first object presented in a before and after image of the object as rotations are applied. The second is shown as a before, with five possible outcomes of the object after applying the same rotation as the example object

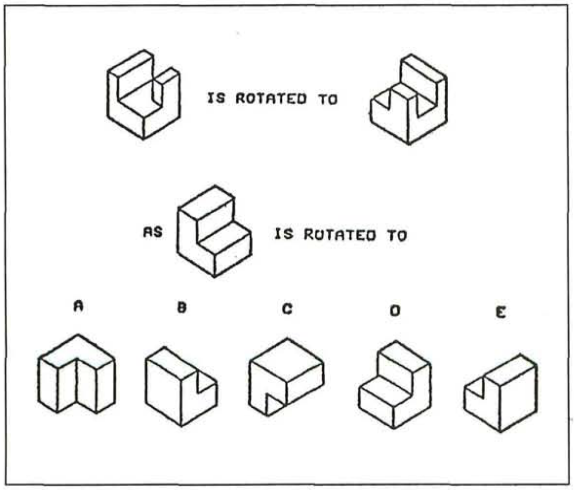

Figure 1: Sample Problem from the PSVT:R

[32]. The third instrument used was a validated revised version of Parker's et al.'s Second Computer Science 1 Exam (SCS1) [18]. We have titled this exam SCS1R. The SCS1R, a subset of the SCS1, consists of nine questions covering topics commonly taught in a first year computer science course and was used to collect students' pre and post programming abilities. Each question is designed to be answerable across different programming languages [3]. The fourth survey used was Dorn and Tew's Computing Attitude Survey version 4 (CASv4) [9]. Dorn and Tew's survey consisted of 26, 5-point likert scale, questions to probe students attitudes towards computation. Each question was then grouped into five factors using exploratory factor analysis [9].

- Problem solving - transfer (PS Tran) : Consisted of questions related to a student's ability to see and/or apply connections between concepts and ideas in order to solve problems.
- Personal interest (Per Int): Consisted of questions related to personal interest, motivation, and engagement with computer science.
- Problem solving - strategies (PS Strat) : Consisted of questions that focused on classic problem-solving strategies in computer science, including topics of practice, problem decomposition, and planning prior to writing code.
- Real world connections (RW Conn): Consisted of questions that dealt with the relationship between the real world and computer science.
- Problem solving - fixed mindset (PS FM): Consisted of questions that relate to a student's belief of predetermined fate or learned helplessness within computing.

3.2 Analysis

A D'Agostino and Pearson's test was used to test for normal distribution for each of the variables. The test combines skew and kurtosis to produce an omnibus test of normality [8]. Throughout this paper, we used an p-value ≤ 0.05 to indicate that the results found are significant. The use of a one-way analysis of variance test

Table 1: Loading Matrix of Each of the Five Attitudinal Factors

	PS Tran	Per Int	PS Strat	RW Conn	PS FM
Q1	.70				
Q2	.75				
Q3	.60				
Q15	.50				
Q4		.73			
Q11		.87			
Q13		.92			
Q26		.82			
Q5			.49		
Q7			.54		
Q10			.61		
Q12			.89		
Q16			.66		
Q17			.58		
Q8				.60	
Q9				.78	
Q14				.81	
Q22				.36	
Q6					.66
Q18					.74
Q20					.73
Q21					.89
Q22					.47
Q23					.74
Q24					.89
Q25					.56

(ANOVA) test, or a Kruskal-Wallis non-parametric (if the data was not normally distributed), was used to test for significant differences between groups [26].

3.2.1 *Factor scores.*

To quantify each of the attitudes factors, factor scores were obtained for each of the participants. To calculate students' scores on each of the five attitudes, factors we used a least squares method (Equation 1), where L is the loading matrix (Table 1) obtained from running an exploratory factor analysis and L' is the transpose of the loading matrix. x_{ij} is a student's response to a given question j, and μ is the average response to a given question.

$$Student_i = (LL')^{-1}L'(x_{ij} - \mu_{ij}) \tag{1}$$

3.2.2 *Multinomial Logistic regression.*

Students were classified into three categories; low, middle or high ability, based on their post SCS1R scores. Any student who scored a 2 or below on the post SCS1R was classified as having low ability because the SCS1R only had 9 questions and by random guessing students should have scored a 2. Students who scored greater than 2 and less than or equal to 5 were classified as middle ability because, the average student score on the post SCS1R during the validation process was a 3.5 [3]. The rest were classified as high ability. There was a total of of 19 high, 92 middle and 86 low ability students.

A multinomial logistic regression model (MLR) was used to predict students' programming abilities. A MLR is a multivariate extension of a logistic regression (LR). A LR can be used for discriminant analysis when the data contains both quantitative and categorical variables and is used for predicting between two categories. LR uses a logistic function to predict the probability of a dependent variable, in our case students' programming ability, as a function of the independent variables. If the probability is greater than or equal to 50%, the model classifies the observation as group 1, else it classifies the observation as group 2. When the data has more than two classes, as it is with our case, a MLR can be used.

A confusion matrix was used to show the results of the model prediction versus the actual classification of the dependent variable. To interpret a confusion matrix, the left most column is what the model classified each student's post programming ability, while the top of the table is the student's actual programming ability. The sum of the diagonal of the matrix is the number of correct times the model categorized a student's ability, while the off diagonals is the number of times the model miss-categorizing a student's ability. Total accuracy is calculated by taking the total number of correct response divided by the total number of observations.

Cross validation was used to evaluate the fit of the model. The idea of cross validation is to simulate running a secondary study to validate the fit of the model. This is done by randomly shuffling the sample data and splitting it up into two data sets. One set is used to build/train our model and a second set is used to test how well that model actually did at predicting students' programming abilities. Our data was split 50/50, where the training set consisted of 98 random observations and the testing set consisted of the remaining 99 random observations.

4 RESULTS

4.1 Normality

A D'Agostino and Pearson's test was used to test if each variable was normally distributed. Students' prior spatial abilities were normally distributed with an f score of 15.6 and a p-value < 0.01. The PS-Transfer attitude factor was not normally distributed with an f score of 5.5 and a p-value < 0.06. The Personal Interest attitude factor was normally distributed with an f score of 20.6 and a p-value < 0.01. The PS-Strategies attitudes factor was not normally distributed with an f score of 0.14 and a p-value = 0.93. The Real World Connections attitude factor was also not normally distributed with an f score of 3.42 and a p-value = 0.18. The last attitude factor, PS-Fixed Mindset, was normally distributed with an f score of 7.94 and a p-value = 0.02.

4.2 ANOVA

After determining what variables were normally distributed, an ANOVA or Kruskal-Wallis test (if the variable was not normally distributed) was used to test for significant differences between groups. There was a significant difference both between students' prior programming abilities (f = 51.3, p < 0.01) and spatial abilities (f = 29.3, p < 0.01) to their post programming abilities. There was also a significant difference between the PS-Transfer (f = 16.91, p < 0.01), Personal Interest (f = 6.05, p = 0.02), and PS-Fixed Mindset (f = 28.54, p < 0.01) attitudinal factors and post programming abilities.

Table 2: Individual Variable MLR Accuracy

	Num Correct	Num Incorrect	Accuracy (%)
Pre-SCS1R	52	47	52.2
Pre-PSVT:R	56	43	**56.6**
Gender	50	49	50.5
SES	44	55	44.4
Race	47	52	47.5
PS-Transfer	47	52	47.5
Personal Interest	45	54	45.5
PS-Strategies	42	57	42.4
Real-World Con.	42	57	42.4
PS-Fixed Mindset	50	49	50.5

There was no significant difference with the PS-Strategies (f = 2.15, p = 0.14) and the Real World Connections (f = 0.02, p = 0.903) attitudinal factors and post programming abilities. As for the categorical variables, there was a significant difference with Gender (f = 6.28, p < 0.01), but not with Race (f = 1.86, p = 0.08), SES (f = 1.25, p = 0.30) and post programming abilities.

4.3 Multinomial Logistic Regression

To answer our first research question, the accuracy of each of the individual variables at predicting students' programming abilities at the end of the semester, an MLR was used on each of the individual variables (prior programming abilities, prior spatial skills, gender, race, SES, and the five attitudinal questions) that were collected at the beginning of the semester to see how well they predicted students' post-programming ability (summarized results found in Table 2). Running an MLR on prior programming abilities resulted in a model that correctly classified 52 out of 99 students, total accuracy = 52.2% . Students' prior spatial skills resulted in a model that correctly classified 56 out of 99 students, total accuracy = 56.6%, the highest out of all the variables. Gender had an overall accuracy of a 50.5%, SES 44.4%, Race 47.5%, PS-Transfer 47.5%, Personal Interest 45.5%, PS-Strategies 42.4%, Peal-World Connects 42.4%, and PS-Fixed Mindset had an overall accuracy of 50.5%.

When simultaneously considering all variables the MLR model categorized a total of 58 students correctly and miscategorized a total of 41 students, an overall accuracy of 58.6%. Results are shown in Table 3. The model did not correctly predicted any students who had high programming abilities; it classified all high programming ability students as having middle programming abilities. The majority of the miscategorizations happened when the model predicted students as having middle programming abilities when they actually had low programming abilities, a total of 24.

A MLR was then ran using the variables that were found to have statistically significant differences based on their f scores obtained by running an ANOVA. Those that were not statistically significant were excluded in he model. Excluding race, SES, PS-strategies and real world connections resulted in a model that categorized a total of 55 students correctly and miscategorized a total of 44 students (Table 4), an overall accuracy of 55.6%.

To find the model that had the highest accuracy, a MLR model was ran on all possible combinations for each variable. It was

Table 3: Model Fit on all Variables (58.6% Accuracy)

Predicted	Actual High	Actual Middle	Actual Low
High	0	0	0
Middle	9	38	24
Low	0	8	20

Table 4: Model Fit on Variables That Had Significant F-Scores (55.6% Accuracy)

Predicted	Actual High	Actual Middle	Actual Low
High	2	0	1
Middle	7	30	22
Low	0	14	23

Table 5: Maximum Model Fit (64.6% Accuracy)

Predicted	Actual High	Actual Middle	Actual Low
High	5	0	6
Middle	2	33	14
Low	2	11	26

found that prior programming abilities, prior spatial skills, SES, PS-transfer, personal interest and PS-fixed mindset resulted in a model that had the highest accuracy. Gender, race, PS-strategies and real world connections added no value or even hurt the overall accuracy of the model. When excluding those variables, the model categorized a total of 64 students correctly and miscategorized a total of 35 students an overall accuracy of 64.6%. This is shown in Table 5.

5 DISCUSSION

It makes sense that students' prior programming abilities should be a good predictor of their post programming abilities. Students who already have high prior programming ability should have high post programming abilities, but students who have low prior programming abilities could either have high, middle, or low post programming ability. What is more interesting is that students' prior spatial skills had the highest accuracy at predicting students' post programming skills. Spatial skills have been shown to be a good predictor in STEM success before, but has never previously been shown to be a predictor in computing.

When using a MLR model on all variables, the model only classified students as having middle or low abilities. This could be due to the fact that less than 10% of students were classified as having high programming ability. However, it did classify all students who had high abilities as having middle abilities and if the goal of this model was to predict at risk students, students who scored 2 or below on the post SCS1R, those students would not have been incorrectly labeled.

When comparing a model that uses all variables to a model that removes a few of those variables, it make sense to see decreased accuracy. What is more surprising is that the model that had the highest accuracy was obtained when gender, race, PS-strategies, and real world connections were removed from the model. While gender had a statistically significant f-value when running an ANOVA and one of the relatively higher individual predictability accuracies, 50.5%, it ended up decreasing the overall accuracy of the model. While SES did not have a significant f-value and had one of the lowest individual accuracies, 44.4%, it remained in the final model.

PS - transfer, personal interest and PS - fixed mindset were all shown to have significant differences when looking at students' post programming abilities. Each of them contributed to the overall accuracy of the model even though they had relatively low individual predictability accuracy. This suggests that students' attitudes towards computing topics prior to taking CS1 course can impact their overall performance in the class. PS-strategies and real world connections did not have significant differences, had the lowest individual predictability accuracy and did not contribute to the final model.

6 LIMITATIONS

The most prominent factor that limited our study was that all participation was voluntary. This could have lead to students not taking the surveys seriously. Students who do not take the surveys seriously will add error to the predictive model, and could be one of the factors that lead to our model not having a high accuracy. It was challenging to label any students who did not take the study seriously because the instrument we used to gauge students' programming ability was difficult and the average student score was a 3.5 out of 9, slightly above random guessing.

There is a second problem with voluntary participating. The subset of students' who chose to participate in the study may not be representative of undergraduates who take an introductory computing course.

The third most prominent factor was that the study was not set up to predict students performance. It was initially set up to gather baseline data to compare what happens to students that go through a spatial skills intervention.

Another limiting factor was that we collected post data from students after the initial drop deadline. We therefore excluded data from students who might have dropped out of the course.

7 CONCLUSION

Our model was able to reach a 64.6% accuracy when accounting for students' prior programming abilities, prior spatial skills, SES, PS-transfer, personal interest, and PS-fixed mindset. We found that when looking at the individual power of each variable, students' prior spatial skills had the highest accuracy when predicting students' post programming performance. These results further confirm that spatial skills are not only a good predictor in engineering disciplines but also in computing disciplines. We believe that we can further improve existing models that predict students' post programming abilities by considering students' prior spatial skills.

ACKNOWLEDGEMENT

This material is based upon work partially supported by the National Science Foundation under Grant Numbers 1711780, 1711830, and 1712331. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Thank you B. McNichol for insightful feedback and comments on earlier drafts of this paper.

REFERENCES

[1] H-D Barker and T. Engida. 2001. Structural chemistry and spatial ability in different cultures. *Chemistry Education: Research and Practice in Europe* 2, 3 (2001), 227–239.

[2] R. Bockmon, S. Cooper, , W. Koperski, J. Gratch, S. Sorby, and M. Dorodchi. 2020. A CS1 spatial skills intervention and the impact on introductory programming abilities. *Proceedings of the 51th ACM Technical Symposium on Computer Science Education (SIGCSE '20)* (2020).

[3] R. Bockmon, S. Cooper, J. Gratch, and M. Dorodchi. 2019. (Re)Validating Cognitive Introductory Computing Instruments. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (2019), 552–557.

[4] P. Byrne and G. Lyons. 2001. The Effect of Student Attributes on Success in Programming. *ACM SIGCSE Bulletin* 33, 3 (2001), 49–52.

[5] C. S. Carter, M. A. LaRussa, and G. M. Bodner. 1987. A study of two measures of spatial ability as predictors of success in different levels of general chemistry. *Journal of Research in Science Teaching* 24 (1987), 645–657.

[6] S. Cooper, K. Wang, I. Maya, and S. Sorby. 2015. Spatial skill training in introductory computing. *ICER '15* (2015), 13–20.

[7] A. Cox, M. Fisher, and P. O'Brien. 2005. Theoretical considerations on navigating codespace with spatial cognition. *17th Workshop of the Psychology of Programming Interest Group* (2005).

[8] R. D'Agostino and E. S. Pearson. 1973. Tests for departure from normality. *Biometrika,* 60 (1973), 613–622.

[9] B. Dorn and A. Tew. 2015. Empirical validation and application of the computing attitudes survey. *Computer Science Education* 25, 1 (2015), 1–36.

[10] S. Fincher, B. Baker, I. Box, Q. Cutts, M. de Raadt, P. Haden, J. Hamer, M. Hamilton, R. Lister, and M. Petre. [n. d.]. Programmed to succeed? A multi-national, multi-institutional study of introductory programming courses. *University of Kent, Computing Laboratory Technical Report 1-05* ([n. d.]).

[11] M. Fisher, A. Cox, and L. Zhao. 2006. Using sex differences to link spatial cognition and program comprehension. *In Proceedings of the 22nd IEEE International Conference on Software Maintenance (ICSM)* (2006), 289–298.

[12] S. Jones and G. Burnett. 2007. Spatial skills and navigatoin of source code. *Proceedings of ITiCSE 2007* (2007), 231–235.

[13] S. Jones and G. Burnett. 2008. Spatial skills and learning to program. *Human Technology* 4, 1 (2008), 47–61.

[14] 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.

[15] S. N. Liao, D. Singaro, C. Alvarado, Griswold W. G, and L. Porter. 2019. Exploring the value of different data sources for predicting student performance in multiple CS courses. *SIGCSE' 19* (2019), 112–118.

[16] R. E. Mayer, J. L. Dyck, and W. Vilberg. 1986. Learning to program and learning to think: What's the connection? *Commun. ACM* 29 (1986), 605–610.

[17] K. Norman. 1994. Spatial visualization-a gateway to computer-based technology. *Journal of Special Education Technology* 12, 3 (1994), 195–206.

[18] M. Parker, M. Guzdial, and S. Engleman. 2016. Replication, validation, and use of a language independent cs1 knowledge assessment. *ICER '16* (2016), 93–101.

[19] M. Parker, A. Solomon, B. Pritchett, D. Illingworth, L. Margulieux, and M. Guzdial. 2018. Socioeconomic status and computer science achievement. *ICER '18* (2018), 97–105.

[20] J. Parkinson and Q. Cutts. 2018. Investigating the relationship between spatial skills and compute science. *ICER '18* (2018), 106–114.

[21] N. Rountree, J. Rountree, and A. Robins. 2002. Predictors of Success and Failure in a CS1 Course. *ACM SIGCSE Bulletin* 34, 4 (2002), 121–124.

[22] D. L. Shea, D. Lubinski, and P. B. Camilla. 2001. Importance of assessing spatial ability in intellectually talented young adolescents: a 20-year longitudinal study. *Journal of Educational Psychology* 93, 3 (2001), 604–614.

[23] I. M. Smith. 1964. *Spatial ability-its educational and social significance.* London : University of London.

[24] S. A. Sorby. 2000. Spatial abilities and their relationship to effective learning of 3-d modeling software. *Engineering Desing Graphics Journal* 64, 3 (2000), 30–35.

[25] K. J. Vicente, B. C. Hayes, and R. C. Williges. 1987. Assaying and isolating individual differences in searching a hierarchical file system. *Human Factors* 29 (1987), 117–124.

[26] R. L. Wasserstein and N. A. Lazar. 2016. The ASA's Statement on p-Values: Context, Process, and Purpose. *The American Statistician* 70, 2 (2016), 129–133.

[27] C. Watson and F. W. Li. [n. d.]. Failure rates in introductory programming revisited. ([n. d.]).

[28] N. Webb. 1984. Microcomputer learning in small groups: Cognitive requirements and group processes. *Journal of Educational Psychology* 76 (1984), 1076–1088.

[29] L. H. Werth. 1986. Predicting student performance ni a beginning computer science class. *Proceedings of the seventeenth SIGCSE technical symposium on Computer science education* (1986), 138–143.

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

[31] B. C. Wilson and S. Shrock. 2001. Contributing to success in an introductory computer science course: a study of twelve factors. *SIGCSE 2001* (2001), 184–188.

[32] S. Y. Yoon. 2011. Revised Purdue Spatial Visualization Test: Visualization of rotations (Revised PSVT:R) [Psychometric Instrument]. (2011).