The Marriage of Mathematics and Programming

Bernardo M. Ferreira[1]*, Lucas M. Souza[1]†, Laira A. Silva[1]‡, Igor M. Félix[1], Leônidas O. Brandão[1], Anarosa A. F. Brandão[2]

[1] Instituto de Matemática e Estatística (IME) - Universidade de São Paulo (USP) - São Paulo - SP
Rua do Matão, 1010 - CEP 05508-090 - São Paulo - SP

[2]Escola Politécnica (EP) - Universidade de São Paulo (USP) - São Paulo - SP

{bernardomartinsf,lucasmens,laira,igormf,leo}@ime.usp.br,anarosa.brandao@usp.br

Abstract. *In a world where computing skills are a must have, programming is still a hard one to acquire and its courses have high failure rates. Numerous studies propose ways to help the early identification of students that might face difficulties and are more likely to fail a programming course. This study aims to find a correlation between their mathematical abilities and grades in an introductory computer science class following the method of a previous study that had promising results. Students were asked to answer a mathematical abilities test and had their results compared to their grades during the course using the Wilconxon-Mann-Whitney non-parametric test. The results indicate a strong correlation between mathematical abilities and programming potential.*

1. Introduction

Computer has greatly changed society leading the world into a place in which having computing skills is no longer a differential, but a need. This massive change has led to a growth in computer science courses enrollment. According to the Taulbee survey, 2018 was the eleventh consecutive year with an increase in the number of new undergraduate computing majors [Association 2019, p.21].

The growth in enrollment is now one of the great changes in computing education and is leaving institutions struggling to offer sections to new students [Bruce 2018]. At the same time that computing teaching is suffering from over-enrollment, it suffers a contradictory problem of high dropout rates in introductory computer science courses (CS1) [Bruce 2018]. One of the reasons for this high dropout rate lies in high failure rates in these CS1 courses.

Several reports in the literature points to students' high failure rates in CS1 courses. In 2019 a research with 161 high education institutions reports found that CS1 courses have an average failure rate of 28% [Bennedsen and Caspersen 2019]. This average is fortunately lower than the one from 2007, with 33% in US institutions and 41% in non US institutions [Caspersen 2007], but still high, specially considering that some reports do not consider students that dropped out mid course [Bennedsen and Caspersen 2019].

*CAPES PROEX Scholarship holder
†CNPq Scholarship holder, 168268/2018-3
‡CAPES PROEX Scholarship holder

DOI: 10.5753/cbie.sbie.2019.1790

In 2006 Wing defined Computational Thinking (CT) and caught the attention of the community leading researchers to study the importance of teaching computational concepts on all educational levels [Grover and Pea 2013]. The definition given by Wing broadens the application of computer science competences not only to programming, but as skills for everyone [Souza et al. 2019].

The skills that are part of CT such as abstraction and decomposition [Wing 2006] are essential in the process of learning how to program. Considering that, Souza et al researched researched a way to identify students that have higher probability to face difficulties when learning how to program [Souza et al. 2019]. Even though was not conclusive, due to the low amount of subjects [Souza et al. 2019], it still showed promising results.

The study presented on this paper aims to find if there is a relation between mathematical skills and programming learning by applying the method described by Souza et al to a larger set of students.

The rest of this paper is presented as follows: section 2 presents background topics and related works, section 3 presents the methods and their applications, section 4 presents the results of the experiments, section 5 presents the consequences of the results and section 6 presents conclusion and future works.

2. Background

2.1. Computational Thinking

One of the first authors to identify programming as a needed skill was Papert in his effort to introduce computer science concepts as a useful skill for children. [Papert 1980].

In 2006 Wing first defined Computational thinking (CT) as a fundamental skill for everyone. CT uses computation fundamental concepts to solve problems, design systems and understand human behavior [Wing 2006]. Aho simplified the definition of CT as ”the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms” [Aho 2012].

In the process of learning how to program it is essential to have a firm grasp on the concepts involved on CT. In a review of the literature Hsu, Chang and Hung identified 19 thinking steps related to CT reported by researchers, among them were abstraction, algorithm design, decomposition and problem solving [Hsu et al. 2018]. These cited abilities are not exclusive to computer science as they are also a part of STEM’s subjects such as mathematics and engineering.

2.2. Why is learning how to program difficult?

Many researchers have studied possible factors that may predict a persons aptitude towards programming. Caspersen lists 25 studies and the abilities they researched and ”the one finding that seems to be most consistent across various investigations -although not strong- is correlation between mathematics score in high school and performance in CS1” [Caspersen 2007].

Caspersen tried to replicate the studies and found that from all the studied skills, mathematics and the student’s effort had the most relevance in the students final grade. The former accounted for 15% of the variance and the latter for 7%.

Souza et al applied a mathematical abilities test involving abstraction, logical thinking, decomposition and algorithm design and found a p-value of 0.07 with a significance level of 0.05 [Souza et al. 2019]. Even though the results were not conclusive, the low quantity of students participating on the experiment may have affected their data.

2.3. Wilcoxon-Mann-Whitney test

The Wilcoxon-Mann-Whitney (WMW) non-parametric test calculates if two independent groups are homogeneous and have the same distribution. Its null Hypothesis (H_0) stipulates that two independent groups have the same distribution [Nachar et al. 2008].

The WMW method equations ”can be understood as the number of times observations in one sample precede or follow observations in the other sample when all the scores from one gruoup are placed in ascending order” [Nachar et al. 2008].

3. Research

This research was carried out with students enrolled in an *online* CS1 course available only to engineering students who had previously failed a *classroom* CS1 course at Escola Politécnica - USP. Since the participants were not randomly chosen this study is categorized as a quasi-experiment.

This study follows the method proposed by Souza et al that had promising results, even though they were not conclusive [Souza et al. 2019].

3.1. Participants

The quasi-experiment was conducted on a group of 118 students participating in an online CS1 course administrated through an online learning management system. All students enrolled on this course had already failed a previous classroom CS1 course, so they all had previous experience with algorithms and programming.

3.2. Methods

In this quasi-experiment, at the beginning of the course the students were asked to answer a mathematical ability test (MAT) and later these results were compared to their grades in the exams. The steps taken in this study are as follows:

1) MAT: A five question test was applied to evaluate their mathematical abilities. These abilities were: abstraction, decomposition, problem solving and algorithm design.
2) Curriculum analysis: In this step the MAT results were compared to two of the three exams that compose the students final grade in the course using the WMW non-parametric test.

3.3. Research Questions

The research questions of this study are stated as follows:

RQ1) Do the MAT grade and exam grades follow the same distribution?
RQ2) Can the MAT results be used as an indicative of the students' programming learning potential?

3.4. Data collection

The MAT was made available at the first week of the course through the same online learning environment that was used during the course. From all the enrolled students (118) only 62 (52.5%) both answered the test and took part in the first exam.

The questions followed the weights given by [Souza et al. 2019] ranging from 1 to 5 (easier to harder) and were designed in order to identify specific abilities. Each question subject and relative weight is presented in table 1 and the full test can be seen in section 3.5.

The test results were calculated using weighted average considering the relative weight of each question and then compared to the first exam results using the WMW non parametric test. The WMW method aims to determine if the sample distribution is even or uneven for the significance level $\alpha = 0.05$. The method used also consider two hypotheses: the null hypothesis (H_0) test = exam and the alternative hypothesis (H_1) test $\neq$ exam.

By the WMW method if the result (p-value) is less then or equal to the significance level (0.05) then the null hypothesis is confirmed, otherwise it is rejected.

Table 1. Weights and subjects of the MAT questions

Question	Relative weight	Subject
Q1	1.0	Logical thinking
Q2	2.0	Decomposition ability and problem solving
Q3	4.0	Mathematical induction and abstraction
Q4	2.0	Logical thinking
Q5	5.0	Abstraction and algorithm design

3.5. Test questions

The questions presented in the MAT are as follows:

Q1. In a soccer championship, the teams score 5 points if they win, 2 if it is a tie and 0 if they lose. With only the last round left, the teams of Professors (P), Masters students (M) and Phd students (Ph) have 38, 37 and 36 points, respectively, while the fourth place, occupied by Undergraduates (U), have less than 30 points. The last round will be:

M x U and P x Ph

About the described situation, consider the affirmations below made by the crowd:

I) If there is a winner in the match P x Ph, the winner will be the champion.
II) In order for the team M to be the champion, they just have to win their match.
III) P is the only team that, even ending the last match in a tie, can still be the champion.

Choose the only answer below that indicates all the correct affirmations made by the crowd:

a) I.
b) I and II.

c) II.
d) III.
e) None of the above.

Q2. Observe the left pyramid below to discover what was the rule applied to build it. Then, deduce the values of A, B, C and D so that the right pyramid follows the same building rule.

100			
65	35		
42	23	12	
27	15	8	4

137			
A	73		
29	35	B	
12	C	D	20

Q3. In a virtual magic land, your family is going to make a surprise birthday party to your grandma and you are in charge to take 2 cakes to the party. Unfortunately, in the way from the bakery to your grandma's house there are 5 bridges and under each bridge there is a troll charging a toll from people trying to pass. The toll you must pay in each bridge is to give half the cakes you're carrying to the troll controlling the bridge.

a) With how many cakes must you leave the bakery in order to get to your grandma's house with exactly 2 cakes?
b) If you had to pass n bridges, with $n \geq 0$, with how many cakes should you leave the bakery?

Q4. Bob only says the truth on Mondays, Wednesdays and Fridays, and always lies in the other days of the week. Today he said: "Tomorrow i will only tell the truth". What day is today?

a) Tuesday.
b) Thursday
c) Friday.
d) Saturday.
e) Sunday.

Q5. Supposing the 8 values below are in a array named Arr (going from Arr[1] to Arr[8]), describe the steps you should follow in order to sort it in ascending order, considering:

I) You are only able to compare and eventually trade the values 2 positions at a time;
II) Your algorithm must work with any values, to which you do not need to know previously.

3	5	0	1	9	2	7	8

4. Results

The MAT was made available during the first week of the CS1 course. From the 118 students enrolled on the course, the analysis was made based of the 62 that fully answered the test and took part on the first exam (E1) and the 41 who took part on the second exam (E2).

As presented in table 2, two analysis were made in each of the exams, one considering all five questions from the MAT and other without the fifth question (the reasoning for the second analysis is explained in the following section).

Table 2. P-value table

Test questions	E1 p-value	E2 p-value	Souza p-value	Significance Level
Q1-Q5	0.005	0.044	0.07	0.05
Q1-Q4	0.007	0.018	0.455	0.05

4.1. Exam 1

In this exam, the p-value was 0.005 confirming the null hypothesis, meaning that the test and E1 grades have similar distributions.

As can be seen in table 3, the fifth question average grade is significantly lower then the others, only 0.39. A posterior analysis was conducted to investigate the reason and it was discovered that 22 students, 35% of the analyzed data, either got the fifth question wrong or did not even try to answer it.

Table 3. Average grades per question (range 0-1)

Test questions	Average grade
Q1	0.86
Q2	0.96
Q3	0.72
Q4	0.90
Q5	0.39

In figure 1 is easy to see that Q5 had the lowest quantity of students that got full points, only 18 students. This is less then half the amount of students that got Q3 right, the second lowest with 41 students.

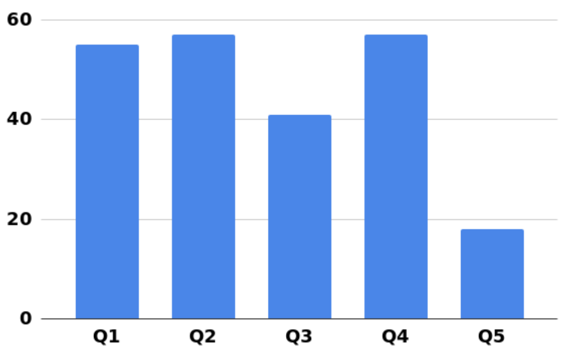

Figure 1. Quantity of students that got full points for each question

Therefore, due to the fact that the fifth question was not only the hardest and most time consuming but also the question most students failed to answer, another analysis was made without it.

Applying the WMW method without the fifth question, the p-value was 0.007 still confirming the null hypothesis.

4.2. Exam 2

From the original 62 students analyzed in E1, 21 (33%) were absent in E2 leaving a total of 41 grades to analyze.

Repeating the previous method, when analyzed with all five questions the p-value was 0.044 and without the fifth question the p-value was 0.018, also confirming the null hypothesis.

5. Discussion

The p-values found in the four analyzed situations validate the null hypothesis, confirming the correlation between the MAT and the exams taken during the course, especially in E1 which had a strong p-value with and without the fifth question.

Considering E1's p-values, the change made by the fifth question was very marginal, differing from E2 in which the p-value with the question is close to 2.5 times larger than without it. The researchers attribute the increased difference to three possible explanations:

1. The test may only represent the learning potential for the most basic programming skills and does not relate so well to more complicated skills such as array manipulation and functions.
2. The high abstention rate of E2 may have impacted the results, as 21 out of the 62 students previously analyzed on E1 were not present. Some reported being absent due to personal reasons and others had already dropped out of the course given their already low grades or time management issues.
3. The second exam was considered very hard by the students as well as the professors. This higher difficulty can be seen in their average grade - while in E1 the students had an average grade of 7.89, in E2 it decreased to 5.22.

Also, this test was made available on the first week of the course while many students still had not logged in the learning management system. Some students had yet to enroll on the course or did not know how to log in the system and three of the students, for example, have never logged in the system. This situation had an impact on the total amount of respondents, only 62 from a total of 118 students.

Although this research used a different method the results contribute and expand Caspersen findings that mathematics is correlated to programming learning potential [Caspersen 2007] by using the same methods applied by Souza [Souza et al. 2019].

6. Conclusion and future work

The results presented in this paper answer positively to RQ1 showing there is a correlation between the students' grades in the MAT and their exams during the course. The results also answer positively to RQ2 as this correlation validates the use of a MAT as a predicative to the students' programming learning potential. This test can be used as a tool to help early identification of students that are more likely to face difficulties, helping the teacher in the usage of preventive measures to diminish the high failure rate in CS1 courses.

Future works devised from this study are:

1. Finish the analysis of this class by analyzing the third exam;
2. Improve the test's questions and include more questions to avoid students guessing answers as well as increasing granularity on the grades;
3. Devise strategies to improve the respondent rate given that, in this work and at [Souza et al. 2019], less then half the class answered the test;
4. Devise strategies to improve students programming learning to minimize the failure rates, for instance, using visual programming and reinforcing basic algorithmic thinking skills.

7. Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001

References

Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7):832–835.

Association, C. R. (2019). 2018 taulbee survey. Technical report.

Bennedsen, J. and Caspersen, M. E. (2019). Failure rates in introductory programming: 12 years later. *ACM Inroads*, 10(2):30–36.

Bruce, K. B. (2018). Five big open questions in computing education. *ACM Inroads*, 9(4):77–80.

Caspersen, M. E. (2007). *Educating Novices in The Skills of Programming*. PhD thesis, Faculty of Science of the University of Aarhus.

Grover, S. and Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational researcher*, 42(1):38–43.

Hsu, T.-C., Chang, S.-C., and Hung, Y.-T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126:296–310.

Nachar, N. et al. (2008). The mann-whitney u: A test for assessing whether two independent samples come from the same distribution. *Tutorials in quantitative Methods for Psychology*, 4(1):13–20.

Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.

Souza, L., Ferreira, B., Brandão, A., Felix, I., Alves Pereira, P., and Brandão, L. (2019). Mathematics and programming: marriage or divorce?

Wing, J. M. (2006). Computational thinking. *CACM*, 49(3):33–35.