ASEE'S VIRTUAL CONFERENCE
At Home with Engineering Education
JUNE 22 - 26, 2020 #ASEEVC
Paper ID #29844

# Evaluating the Effectiveness of Lab Practice in Context of Prior Programming Experience in an Introductory Programming Course

**Caleb James O'Malley, University of Florida**

Caleb O'Malley is a third-year undergraduate student of Computer Engineering at the University of Florida. Caleb has been a research assistant with the Computational Reasoning Group at UF for over a year, studying different aspects of computer science education for non-majors.

**Ashish Aggarwal, University of Florida**

Ashish Aggarwal is a Lecturer of Computer Science in the Department of Engineering Education at the Herbert Wertheim College of Engineering, University of Florida. His research focuses on Computer Science Education and Learning Analytics where he studies the effectiveness of different learning approaches on students' learning outcomes and performance in introductory programming courses.

# Evaluating the Effectiveness of Lab Practice in Context of Prior Programming Experience in an Introductory Programming Course

## Abstract

Understanding major factors, which influence students' learning outcomes in introductory programming (CS1) courses, enables educators to make more effective course designs and provide more valuable learning experiences. Factors like gender and prior programming experience have been studied and linked to students' success, but there is little an instructor can do to change them. As the enrollments continue to increase and diversify, it will be useful for instructors to know what type of course design is effective for which category of students. The lab component of the course provides additional practice to students, but less is known about the usefulness of the lab practice in students' success. We are interested in understanding the effectiveness of lab practice in improving a students' learning outcomes in CS1 courses, especially in the context of prior programming experience. We present the analysis from a CS1 course for non-majors where 82 students were enrolled in a two-credit course, out of which 29 were also enrolled in a one-credit programming lab course. We find that overall lab enrollment is a factor affecting student performance. However, students with little to no prior programming experience benefit significantly more from the lab, compared to students with substantial programming experience. This effect is further found to be concentrated towards the first half of the course, implying a greater importance of the lab in the first few weeks of the course. These results will help guide instructors in tailoring the course design to meet the needs of students with varying prior programming experiences.

## 1. Background

Introductory programming courses have a reputation for being difficult and continue to experience consistent high dropout rates [1], [2]. If instructors are able to better understand which factors contribute to a student's success in their course, especially at the start of the course, better support can be provided to struggling students. It is common for introductory courses to have very high student to instructor ratios, which makes the task of ensuring students' success very difficult. In addition, instructors often do not have a good understanding of how individual students are doing in the course until after the first half of the course [3]. In this setup, struggling students usually are not identified until several weeks after the course has begun, when it is often too late or extremely difficult to provide effective support [3]. Many researchers have studied various factors for their ability to influence the performance of a student in an introductory programming course discussed below.

### 1.1 Factors of Success

A wide range of factors spanning from a student's gender to their experience with video games have been studied in the context of student success in programming courses. Some of the most commonly analyzed factors include gender [3], [4], [5], [6], prior programming experience [3], [5] – [9], and previous math or science courses [3], [8]. Other factors include self efficacy [6], [8], comfort level [3], [6], [10], motivation [10], and attributions [6], [8].

There is currently little evidence that gender plays a major role in student success. Quille et al. [4] conducted a multi-institutional study and found that female students on average had higher previous math course grades and end of year pass rates, as well as lower dropout rates. Despite this, female students were found to have lower self efficacy, lower expected end-of-year grade and higher anxiety during tests. In addition, Bergin et al. [3] noted some differences between male and female students when assessing the predictive ability of fifteen different factors.

Studies that include prior programming experience as a possible factor of student success have found mixed results. Some studies [5], [8], [9] conclude that prior experience is correlated with student success. Others find a weak or non-existent relationship between the two [3], [6].

A comprehensive study by Wilson et al. [6] found comfort level to be the most important factor of student success among the twelve they tested. However, Bergin et al. [10] found that intrinsic motivation and high self efficacy were more essential than comfort level in contributing to students' success.

A study from Watson et al. [8] compared the effectiveness of 38 traditional predictors with 12 new data-oriented ones which take into account a student's specific programming behaviors. Among all 38 traditional factors studied, only self efficacy and attribution of success to ability were found to be strong predictors of student success. All other factors found to be strong predictors of success were in the new data-driven category.

**1.2 Possible Solutions**

Overall, there is much variation in the results of these studies. Instructors may find it useful to understand which factors likely contribute to their students' success, but it is difficult to be certain about what factors are most important. With enrollment populations continuing to increase in both size and diversity, it is difficult to manage support systems for CS1 courses. In addition, most of these factors are determined before the student enters a programming class (prior programming experience, gender, previous course performance etc.). The best an instructor can do is try to target struggling students and provide more practice and support to those who need it. Less is known about the impact of additional practice on student success, but one can hypothesize that more practice will result in better performance. The lab portion of a programming course is one way in which students receive practice on programming exercises. Normally, the lab portion of a programming course is included with the rest of the course, and all students complete the lab exercises. But we don't have an exact way to explore if the lab is useful to the students and who benefits the most from the lab practice and why? We believe it is important to study the effectiveness of lab practice and understand its value in context of which type of students is it useful for so that it can be adaptively or prescriptively used in the courses.

We are interested in studying the following two research questions:

***Research Question 1: Is programming practice in the lab helpful in improving student-learning outcomes?***

***Research Question 2: If so, who is benefitting the most? If not, why is this practice not helpful?***

In this paper, we analyze the effect of lab practice by studying the student performance of those who are enrolled in it and those who are not. The structure of this CS1 course for non-majors offered at a public R1 research institution in the Southeast U.S.A is such that students can choose to enroll in the programming course with or without enrolling in the programming lab. Therefore, we have the opportunity to study these two groups which can be compared through the real course-experience of students. The answers to the following research questions will help instructors to provide information to students, which helps students to make more informed decisions about their lab enrollment. We believe that this will better inform a CS1 instructors' decisions regarding course design, creating an environment which is more tailored to the needs of a large and diverse enrollment population.

## 2. Course Context

This study took place throughout the Summer semester of an introductory programming course for non-majors. Details concerning the structure of each course as well as the demographics of the students are described below.

### 2.1 Structure of the Courses

The students analyzed in this study were able to enroll in a two-credit traditional lecture course, with or without being enrolled in a separate one-credit lab course. This arrangement allowed us to make some interesting observations based on a student's enrollment in one or both of the courses. The individual courses are broken down below.

#### 2.1.1 Traditional Lecture-based CS1 Course

All students involved in the study were enrolled in this two-credit introductory programming course for non-majors using MATLAB. A large portion of all course content and interaction takes place online through the learning management system. For this reason, it is described as a "hybrid course" and an "online-course". Lectures are held in-person twice each week, and attendance is optional. Each lecture is recorded and posted online after the class. Many students choose not to attend lectures in person, and simply watch them online once they are posted. About one-fourth of the students attend lectures in person on any given day.

In addition to watching lectures, students complete one project-based homework assignment each week. These assignments involve reading, writing, solving and reasoning about a mini-project like single problem in MATLAB which are expected to be difficult. Due to their difficulty, students are able to collaborate with other students, attend office hours, and access the internet for help throughout the week.

The other course resources offered are standard among any introductory programming course: practice exams, office hours, and some additional content on the course's online page. The structure of the exams is also fairly standard. About half of the exam involves solving problems by writing out programmatic solutions. The rest of the exam consists of questions based on predicting the output from a given code snippet and debugging the programs. There are only two exams throughout the semester. The first is a midterm exam, covering basic programming concepts like conditionals and loops. The second, which occurs at the end of the semester, covers more advanced topics like vectors, strings, matrices and images and is normally seen as more difficult than the first exam.

### 2.1.2 Programming Lab Course

This one-credit lab course acts as an add-on to the lectures and homework which comprise the first course. Students are given a set of programming exercises to solve and practice in the lab. During the labs, students are free to seek help and discuss with their neighbors and the student teaching assistants who are available with the instructor in the labs. The goal of the labs is to be an additional practice session based on the contents covered in the class. This one-credit lab is optional, as some engineering majors require it and some do not. So, the lab acts like an enforced-practice session for the students and no new concepts are discussed in the lab.

Lab time is also a good time to ask questions. It is not uncommon for students to be shy about coming into office hours, or raising a hand during lecture. By practicing these problems in the presence of student teaching assistants who are available to help, students can get their questions answered on the spot. This system is intended to build students' confidence in their ability to program and clear up common misconceptions that students may have when first learning to program.

## 2.2 Demographics

A total of 97 students were enrolled in the course. A survey was distributed at the end of the semester which asked students for their name, gender identity, and for them to categorize their prior programming experience in terms of time spent programming. The options available for prior experience were listed as follows: "0-10 hours", "10-100 hours", "100-1000 hours", or "I am a software developer". 82 students responded to the survey out of the 97 enrolled. Of the 82 students who did respond to the survey, all were enrolled in the traditional course and 29 out of these 82 students were also enrolled in the lab course. In the analysis presented in this paper, we study these 82 students. The survey was voluntary, and we have no reason to believe that those students who did not respond to the survey will bias the data in any particular way.

From the students' responses, we observed the following:

- By gender, 61% of the students were male and 39% were female.
- By prior programming experience, 56% of students identified with the lowest level of programming experience,
- 34% identified with the second-lowest level of programming experience,
- And the remaining 10% of students identified with greater levels of prior experience.

## 3. Methodology

### 3.1 Data Collection

In addition to the survey data, the course's learning management system (LMS) was used to collect information about the students' exam grades and lab enrollment. All the required approvals were taken before conducting the analysis and data was anonymized.

### 3.2 Analysis Methods

#### 3.2.1 Variables

The goal of this analysis is to understand the effects that a students' prior programming experience, gender, and enrollment in the lab course might have on a student's performance in the course. The factors of prior programming experience (PPE), lab enrollment, and gender are all defined as binary variables during analysis. For prior experience, students that responded "0-10 hours" were given a PPE value of 0, and students who responded with any higher level of experience were assigned a PPE value of 1. Although more than two distinct values of prior experience were able to be reported, very few students reported prior experience greater than the two smallest options. Therefore, this factor was made binary for simplicity. Lab enrollment was quite straightforward as a binary variable (students are either enrolled or not enrolled), and gender was only treated as binary in this study due to the fact that all responses to the survey were either "Male" or "Female". No students selected the alternate options, which allowed for user input.

#### 3.2.2 Measuring Student Performance

When determining how to measure students' performance in the course, we had a few options. Exam score averages, assignment score averages, or final grades in the course could have been used to represent a student's success. It was determined that students' exam scores would be the best measure of success in this course. All homework and project assignments allowed students to seek help from other students, office hours, and the internet. Students also had extended periods of time to work on these assignments. The result of this is that most students earned very high scores on projects and assignments. Therefore, using this data in a formal analysis would not be useful, as there is little variation present in the data set. The same is true for students' final grades, because their final grades are influenced in part by the high homework and project scores just described. The students' exam score, however, provided a data set, which allowed us to measure their understanding under similar conditions which is a strong measure for students' success. We used the average of the students' two exam scores for the majority of analysis. In section five, however, the differences between the two exams is discussed. In addition, exams were designed to be challenging, testing students on program reasoning, debugging and writing. If a student has developed these skills throughout the semester, they should perform well and be considered successful.

### 3.2.3 T-tests

One statistical tool used for analysis is the t-test, which allows us to check for significant differences between each group's mean course performance. When performing the t-tests, the critical t-stat being used to check for significance is two-tailed, checking if either group performed significantly better or worse than the others. Before conducting the t-test, an F-test is carried out to check if the two groups being compared differ in variance. The t-test carried out afterwards either assumes equal variances or unequal variances, according to the result of the previous F-test. All tests use an alpha value of 0.05 to determine significance.

First, t-tests are first performed on groups of students which have only been sorted by a single factor. For example, all male students and all female students are compared to each other as two large groups, but the students within these groups still differ in the dimensions of prior experience and lab enrollment. From these tests, very broad effects can potentially be observed. In this first round of our analysis, the following groups are compared according to their mean exam scores:

- Male students compared to female students
- PPE=0 students compared to PPE=1 students
- Students not enrolled in lab compared to students enrolled in lab

For the second round of analysis, students are divided into subgroups based both on their prior experience and lab enrollment for separate analysis. A t-test is carried out to compare these subgroups, which now only contains students which differ from each other in only one way. For example, students enrolled in the lab with PPE=0 are compared with students who were enrolled in the lab with PPE=1. By filtering students into smaller groups, the number of variables at play is reduced, allowing for more detailed and interesting findings.

### 3.2.5 Tests for Assumptions

In order to understand the validity of our data set and the subsequent analysis, a few assumptions must first be tested which the factorial ANOVA takes into account. The assumption of normality was tested via examination of the residuals. Review of the Shapiro–Wilk (S-W) test for normality and skewness and kurtosis statistics suggest whether or not normality was a reasonable assumption. The boxplot of the residuals is analyzed for a relatively normal distributional shape. The Q–Q plot and histogram are also examined for normality. The assumption of homogeneity of variance (homoscedasticity) is tested by conducting Levene's test.

Random assignment of individuals to groups was not possible in this study due to the nature of the independent variables being studied, so the assumption of independence was not met in this sense. However, scatterplots of residuals against the levels of the independent variables are reviewed to check this assumption further. A random display of points around 0 provide evidence of whether or not this data behaves independently. The results of each of these tests are reported in the following section. All the statistical analysis was conducted in IBM-SPSS 11.

## 4 Findings

All of the following findings were determined using the statistical methods just described. Course performance is represented by the average of each student's two exam scores.

### 4.1.1 Tests for Assumptions

The assumption of normality was tested and met via examination of the residuals. Review of the S–W test for normality (SW=0.987, df=82, p=0.593) and skewness (0.017) and kurtosis (−.559) statistics suggested that normality was a reasonable assumption. The boxplot suggested a relatively normal distributional shape (with no outliers) of the residuals. The Q–Q plot and histogram suggested normality was reasonable as well. According to Levene's test, the homogeneity of variance assumption was not satisfied [F (3, 78)=3.935, p=0.011]. Scatterplots of residuals against the levels of the independent variables were reviewed to check the assumption of independence. A random display of points around 0 provided evidence that the assumption of independence was reasonable.

Table 1: Exam averages and p-values for each group of students.

| Group | Exam Average | p-Value |
|---|---|---|
| Female<br>Male | 80.0<br>77.4 | 0.317 |
| Not in Lab<br>In Lab | 76.2<br>82.4 | 0.012 |
| PPE=0<br>PPE=1 | 76.7<br>80.6 | 0.127 |

### 4.2 Gender

The F-test for variances determined that the variances between male and female exam scores are not significantly different (p=0.176). The t-test (table-1) assuming equal variances did not provide evidence that gender is a significant factor for students' average exam scores (p=0.317). Gender was not examined any further than this initial t-test and within section five.

### 4.3 Prior Programming Experience

The F-test for variances determined that the variances between exam scores of those with PPE=0 and PPE=1 are significantly different (p=0.013). The t-test (table-1) assuming unequal variances did not provide evidence that prior programming experience is a significant factor for students' average exam scores (p=0.127). However, further t-tests provided a more interesting picture.

**4.4 Lab Enrollment**

The F-test for variances determined that the variances of exam scores between students enrolled in lab and not enrolled in lab are significantly different (p=0.011). The t-test (table-1) assuming unequal variances provided evidence that lab enrollment is a significant factor for students' average exam scores (p=0.012).

**4.5 Relating Prior Programming Experience and Lab Enrollment**

For this analysis (table-2), students are separated according to both prior programming experience and lab enrollment. t-tests are performed on groups differing in only one of these dimensions, disregarding gender. Gender is not taken into account during this round of analysis because no relationship has been observed during the first portion of our analysis.

Table 2: Exam averages and p-values for each PPE/Lab combination.

| Groups | | Exam Average | p-Value |
|---|---|---|---|
| PPE=0 | Not in Lab | 72.8 | 0.006 |
| | In Lab | 82.8 | |
| PPE=1 | Not in Lab | 80.1 | 0.626 |
| | In Lab | 81.7 | |
| Not in Lab | PPE=0 | 72.8 | 0.036 |
| | PPE=1 | 80.1 | |
| In Lab | PPE=0 | 82.8 | 0.761 |
| | PPE=1 | 81.7 | |

When PPE=0: Students enrolled in the lab course performed significantly better (p=0.006) than students not enrolled in the lab course.

When PPE=1: Students enrolled in the lab did not perform significantly different (p=0.626) than students not enrolled in lab.

When students are not enrolled in the lab: Students with PPE=1 performed significantly better (p=0.036) than students with PPE=0.

When students are enrolled in the lab: Students with PPE=1 did not perform significantly different (p=0.761) than students with PPE=0.

Overall, students performed better when enrolled in the lab or when entering the course with more prior programming experience.

Table 3: Results of Factorial ANOVA using average exam scores.

Tests of Between-Subjects Effects
Dependent Variable: Exam Avg.

| Source | Type III Sum of Squares | df | Mean Square | F | Significance | Partial Eta Squared | Noncent. Parameter | Observed Power[b] |
|---|---|---|---|---|---|---|---|---|
| Corrected Model | 1402.646[a] | 3 | 467.549 | 3.668 | 0.016 | .124 | 11.003 | .782 |
| Intercept | 453312.761 | 1 | 453312.761 | 3556.083 | < 0.001 | .979 | 3556.083 | 1.000 |
| Lab | 605.204 | 1 | 605.204 | 4.748 | 0.032 | .057 | 4.748 | .576 |
| PPE | 173.899 | 1 | 173.899 | 1.364 | 0.246 | .017 | 1.364 | .211 |
| Lab * PPE | 307.181 | 1 | 307.181 | 2.410 | 0.125 | .030 | 2.410 | .335 |
| Error | 9943.074 | 78 | 127.475 | | | | | |
| Total | 515395.000 | 82 | | | | | | |
| Corrected Total | 11345.720 | 81 | | | | | | |

a. R Squared=0.124 (Adjusted R Squared=0.090)

b. Computed using alpha=0.05

**4.6 Results of Factorial ANOVA**

From Table 3, we see that the interaction of lab enrollment by prior programming experience is not statistically significant, but there is a statistically significant main effect for lab enrollment ($F=4.748$, $df=1, 78$, $p=0.032$). Effect z is very small for enrollment and prior experience (partial $\eta_{lab}^2=0.057$; partial $\eta_{ppe}^2=0.017$), and observed power is moderate for lab enrollment, but not large for prior programming experience (power=0.576 and .211, respectively). Figure 1 visualizes the possible interaction between PPE and lab enrollment using estimated marginal means.

**5. Differences Between Exams**

All previous analysis was performed using the average of the student’s two exam scores. When the same statistical tests are performed on exam 1 and exam 2 individually on single factors, the following findings are observed.

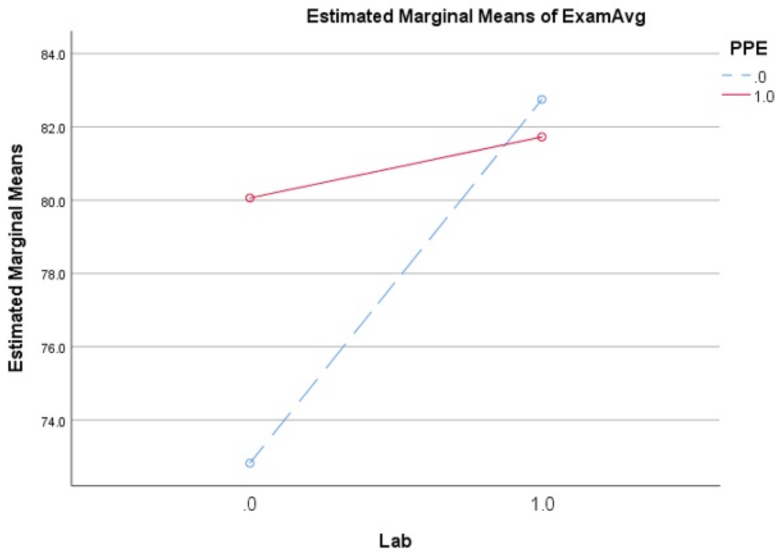


Figure 1: Line plot of EMM's for PPE and Lab, visualizing possible interaction.

**5.1 Gender across Exams**

Exam 1: Male and female students did not perform significantly different from each other (p=0.286).

Exam 2: Male and female students did not perform significantly different from each other (p=0.478).

Overall, gender does not appear to play a major role in student performance.

**5.2 Prior Programming Experience across Exams**

Exam 1: Students with PPE=1 did not perform significantly different from students with PPE=0 (p=0.083). Students with PPE=1 did score more than five points higher on average than students with PPE=0.

Exam 2: Students with PPE=1 did not perform significantly different from students with PPE=0 (p=0.421). Students with PPE=1 scored 2.3 points higher on average than students with PPE=0.

Prior programming experience does not appear to play a major role in course performance when analyzing exams individually.

**5.3 Lab Enrollment across Exams**

Exam 1: Students enrolled in the lab course performed significantly better than students not enrolled in the lab course (p=0.002). Students enrolled in the lab course scored about nine points higher than students not enrolled in the lab course.

Exam 2: Students enrolled in the lab course did not perform significantly different than students not enrolled in the lab course ($p=0.286$). Students enrolled in the lab course scored a few points higher than students not enrolled in the lab course.

Early in the course, students in lab outperform those not in the lab. Over time, these students begin to perform more similarly.

### 5.4 Relating Prior Programming Experience and Lab Enrollment on Individual Exams

The tests just described only involve sorting the students by one factor. By splitting up these groups into subgroups based on prior programming experience as well as lab enrollment, more specific findings emerge.

#### 5.4.1 Relating PPE and Lab Enrollment: Exam 1

When PPE=0: Students enrolled in the lab course performed significantly better ($p=0.001$) than students not enrolled in the lab course.

When PPE=1: Students enrolled in the lab did not perform significantly different ($p=0.393$) than students not enrolled in lab.

When students are not enrolled in the lab: Students with PPE=1 performed significantly better ($p=0.027$) than students with PPE=0.

When students are enrolled in the lab: Students with PPE=1 did not perform significantly different ($p=0.893$) than students with PPE=0.

The lab course appears to benefit students with lower levels of prior programming experience the most early in the course. In addition, prior programming experience is more of a factor for students not enrolled in the lab course early on.

#### 5.4.2 Relating PPE and Lab Enrollment: Exam 2

When PPE=0: Students enrolled in the lab course did not perform significantly different ($p=0.160$) than students not enrolled in the lab course.

When PPE=1: Students enrolled in the lab did not perform significantly different ($p=0.934$) than students not enrolled in lab.

When students are not enrolled in the lab: Students with PPE=1 did not perform significantly different ($p=0.198$) than students with PPE=0.

When students are enrolled in the lab: Students with PPE=1 did not perform significantly different ($p=0.727$) than students with PPE=0.

On exam 2, the trends from exam 1 persist, but to a smaller degree. Differences between students with varying levels of prior experience and lab enrollment are not as apparent as they were in exam 1.

## 6. Discussion

A key finding of this study is the connection between prior programming experience and a student's enrollment in the lab course. Overall upon analyzing students who were enrolled in the lab with those who were not, the students enrolled in the lab course performed significantly better than the students not enrolled in lab. A significant difference between those enrolled in lab and those not enrolled was observed on exam 1, but not on exam 2.

Upon further analyzing we find that students who did have prior programming experience and did not take the lab, had no significant difference in their performance as compared to students who did have prior experience and who were enrolled in the lab. But students who did not have any prior programming experience and were enrolled in the lab performed significantly better than the students who were not enrolled in the lab and who did not have the prior programming experience.

We find that when students are enrolled in the lab course, they perform similarly to each other regardless of their level of prior programming experience. On the contrary, students who are not enrolled in the lab course perform better on average if they have prior programming experience. In short, a student needs to either be enrolled in the lab course or have prior experience in order to have a better chance of succeeding. Both prior programming experience and lab enrollment provide additional practice to students, which in turn increases ability in programming and performance in the course overall.

## 7. Recommendations

The student's ability to choose to take their programming course with or without lab grants the student more options when planning their college experience, allowing for a more effective course load. For example, we find that students with prior programming experience who also enrolled in the lab did not perform much better than those with experience who did not enroll in lab. Therefore, if a student has prior knowledge in programming, they can be advised that the lab probably is not a requirement for success in this course. On the contrary, students with no prior experience can be advised to enroll in the lab course in order to receive the extra help they often need. In general, the student experience becomes more optimized if this information is put to good use in advising scenarios.

We believe that as many courses are adopting the flipped model for CS1 courses, our findings suggest that students with no prior experience are expected to benefit immensely from the in-class activities, while still supporting those with prior experience as well.

## 8. Threats to Validity

Students' self-reporting of prior programming experience is a possible threat to the validity of our results.  In addition, students self report of the number of hours they have spent programming may differ from their actual experience. However, we do not expect that many students reported incorrect data.

Using exam scores to represent the success of the student also has limitations. The student experience is certainly more nuanced than what two numbers can tell us. One bad test day could cause an excellent student to appear to be performing poorly.

## 9. Conclusion

In line with previous works in the area of predictive factors, gender was not found to be a significant factor of course success. Programming experience was also not found to be a crucial factor overall, but did produce some interesting results in combination with lab enrollment. A student's lab enrollment was observed to be the most important factor of a student's success on a broad scale. The lab serves not to give certain students an edge over others, but to level the playing field between those who have prior programming experience and those who don't. Students in the lab performed similarly in the course regardless of their prior programming experience. However, when looking at students who were not in the lab, those with prior experience performed significantly better than those without it. Students can benefit from these ideas when scheduling courses, and make better decisions based on their familiarity with programming.

**10. References**


[1] Jens Bennedsen and Michael E. Caspersen. 2007. Failure rates in introductory programming. SIGCSE Bull. 39, 2 (June 2007), 32-36. DOI: https://doi.org/10.1145/1272848.1272879

[2] T. Jenkins. On the Difficulty of Learning to Program. In Proceedings for the 3rd Annual conference of the LTSN Centre for Information and Computer Sciences , Loughborough, UK August 27 - 29, 2002.

[3] Bergin, Susan & Reilly, Ronan. (2005). Programming: Factors that Influence Success. ACM SIGCSE Bulletin. 37. 411-415. 10.1145/1047124.1047480.

[4] Keith Quille, Natalie Culligan, and Susan Bergin. 2017. Insights on Gender Differences in CS1: A Multi-institutional, Multi-variate Study. In Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17). ACM, New York, NY, USA, 263-268. DOI: https://doi.org/10.1145/3059009.3059048

[5] Kersteen, Z. A., Linn, M. C., Clancy, M., & Hardyck, C. (1988). Previous Experience and the Learning of Computer Programming: The Computer Helps Those Who Help Themselves. Journal of Educational Computing Research, 4(3), 321–333. https://doi.org/10.2190/9LE6-MBXA-JDPG-UG90

[6] Brenda Cantwell Wilson and Sharon Shrock. 2001. Contributing to success in an introductory computer science course: a study of twelve factors. In Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education (SIGCSE '01). ACM, New York, NY, USA, 184-188. DOI: https://doi.org/10.1145/364447.364581

[7] Anya Tafliovich, Jennifer Campbell, and Andrew Petersen. 2013. A student perspective on prior experience in CS1. In Proceeding of the 44th ACM technical symposium on Computer science education (SIGCSE '13). ACM, New York, NY, USA, 239-244. DOI: https://doi.org/10.1145/2445196.2445270

[8] Christopher Watson, Frederick W.B. Li, and Jamie L. Godwin. 2014. No tests required: comparing traditional and dynamic predictors of programming success. In Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14). ACM, New York, NY, USA, 469-474. DOI: https://doi.org/10.1145/2538862.2538930

[9] Christine Alvarado, Cynthia Bailey Lee, and Gary Gillespie. 2014. New CS1 pedagogies and curriculum, the same success factors?. In Proceedings of the 45th ACM technical symposium on Computer science education (SIGCSE '14). ACM, New York, NY, USA, 379-384. DOI: https://doi.org/10.1145/2538862.2538897

[10] Bergin, Susan & Reilly, Ronan. (2019). The influence of motivation and comfort-level on learning to program.