Performance and Attitudes by Prior Programming Experience

Emily Danchik and Edith Hanson
Abstract

Our research aimed to determine if there is a correlation between the programming knowledge a student has upon entering CSC 171 and the final grade he or she earns. To test this, we surveyed the class to find each student's entering knowledge level, and generated statistics comparing knowledge level and average quiz score, and also knowledge level and midterm grade. We found a trend line showing that entering with more knowledge guarantees the student a high grade, while a student entering with no background will most likely earn a lower grade. We propose that a seminar be offered at the beginning of the fall term to teach the students with less preparation about basic programming skills. This will create a knowledge base for them upon which to grow.
Introduction

CSC 171 is billed as an introductory class, requiring no previous knowledge of programming. As such, students from all backgrounds take the course: some have never seen computer code before, whereas others could probably skip the course if they wanted.

Despite having no prerequisite, upon entering the course, new material is presented at a blinding speed. Beginning students often flounder about, feeling overwhelmed with learning the absolute basics of a completely new topic. At the same time, the more seasoned programmers in the class flourish, taking full advantage of the opportunity to start off the semester with excellent marks.

Some of this disconnect is fixed through the workshop program, where students are asked mainly to teach each other the material, and through lab sessions, where teaching assistants are present for the students' completing of homework assignments, available for help at any time. However, the students who enter the class more prepared have a much easier time completing the work.

In an attempt to formalize these thoughts, we chose to research class performance as a function of previous programming knowledge. Our belief is that, by entering the class already knowing a portion of the material, a student will earn a higher grade than one who has never programmed before.

Previous Research

Several studies have taken place attempting to determine factors in beginning computer science course success, but not all studies agree. In 2001, Wilson and Shrock found that previous computer science coursework was not a significant indicator in predicting a student's potential success. However, Wilson claims in a 2002 study that prior knowledge of programming has a
significant role in predicting academic success. This sentiment is echoed in Shashaani's 1997 study, which claims that taking a high school programming course is a sure indicator of future success.

**Methods**

In order to collect data from the students, we asked them to complete a short survey. Along with their names, the students were asked to rank their programming knowledge upon entering the course from one to five, one being minimal knowledge and five being very knowledgeable. With the students' permission, we then looked at each student's midterm grade and average quiz score, creating two sets of data: previous knowledge versus midterm grade, and previous knowledge versus average quiz score. By having two sets of data, we hope to provide a better view as to the students' achievement in the course.

We also asked the students about their attitudes towards the class. First, the students were asked to rank themselves knowledge-wise in comparison to their peers: were they more knowledgeable, less knowledgeable, or at the right level? We had already asked the students to rank their knowledge from one to five, so comparing that distribution to the responses to this question could provide some interesting data as to the average student in the class. In addition, the students were asked how they felt about their responses to the following question: if a student feels like he is far ahead of his classmates, does he feel held back? If he lacked programming know-how coming in, does he feel that he can keep up? The responses to this question will help determine whether or not the class caters to the majority of the students.
Finally, we asked the students if they would be willing to attend a brief seminar at the beginning of the term to become familiar with basic programming concepts. We believe that, although there are no prerequisites, the students are expected to understand how to complete tasks such as choosing the correct text editor, compiling, and running a program. Some students enter not even knowing how to do that, let alone what it means to “compile” something. We believe that, if there is a disconnect at the beginning of the term due to lack of student preparation, such a session would help start the new students off strong.

Results

The students who ranked their programming knowledge at one upon entering the course were the largest sample group, consisting of twenty-five responses, which was almost half of our total responses. This group had an average quiz score of 82.15, with a standard deviation of 17.88 points, and an average midterm grade of 77.06, with a standard deviation of 13.73 points.

Nine students ranked themselves at “two,” with an average quiz score of 85.72, and a standard deviation of 16.87 points. This same group had an average midterm grade of 84.72, with a standard deviation of 7.49 points.

Five students responded as “threes.” These students had an average quiz score of 93.38, with a standard deviation of 4.83 points, and an average midterm grade of 85.1, with a standard deviation of 8.01 points.

Eight students reported their knowledge at level four. This group had an average quiz grade of 94.49, with a standard deviation of 6.13 points, and an average midterm grade of 93, with a standard
deviation of 3.81 points.

The smallest sample group was those who ranked their knowledge at five upon entering the course, consisting of three respondees. Of those, the average quiz score was 87.08, with a standard deviation of 3.14 points, and an average midterm grade of 93.17, with a standard deviation of 7.52 points.

When asked about their attitudes toward the class, about 31% of the students responded that they felt their peers were further ahead of them. Also, 35% of the class feels that their peers know about as much as they do, and 35% feel that their peers know less.
When prompted about their feelings regarding these responses, about 4% of the class felt held back, and about 17% feel left behind. Also, about 42% of class members don't notice being particularly held back or left behind, and about 36% responded that they don't care.

In addition to this, when asked if they would attend a seminar at the beginning of the year to teach basic programming concepts, 62% of the respondees indicated that they would.

Discussion of Results

Our data clearly shows that it is possible for a student with little to no background knowledge to take CSC 171 and earn a high grade. In fact, the largest test group, those with “level one” experience upon entering the course, also had the highest standard deviation among any of the groups, showing that it is entirely possible for a new programmer to achieve, but also to fail.
That being said, the average quiz and midterm scores were higher for more seasoned programmers than new students. The trend line on the graph shows a slight increase in average quiz grades from little to high experience, but a more dramatic difference in midterm grades. Also, the grades for new students were across the chart, while those with previous programming experience cluster around the 90 mark.

Conclusions

Although it is possible for new programming students to achieve, previous programming experience does have a factor in students' performance in CSC 171. For example, the average midterm grade for well-prepared students was a 93, while the new students' averaged a 77.06. Since this is a freshman introductory class with no prerequisites, we believe that steps should be taken to close this gap slightly.

When asked about their knowledge in relation to their peers, almost exactly a third, a third, and a third of the class were in each of the three categories (peers ahead, peers at the same level, peers behind). While this distribution is just about perfect, we believe that any attempt to shift students from the “peers ahead” to the “peers at the same level” group would be welcome.

This can be solved, in part, by the seminar at the beginning of the year to teach basic concepts. Over half of the class indicated that they would have gone, so we believe it would be worthwhile for next year’s workshop leaders to consider this possibility. This will provide a foundation for the students' later knowledge, hopefully instilling confidence and willingness to work in them.
References

