AMERICAN SOCIETY FOR ENGINEERING EDUCATION

Paper ID #20936

# You Might (or Might Not) Know More Than You Thought: Student Self-Perception vs. Performance in First Year Engineering Graphics and Programming

**Ms. Natalie C.T. Van Tyne, Virginia Polytechnic Institute and State University**

Natalie Van Tyne is an Associate Professor of Practice at Virginia Polytechnic Institute and State University, where she teaches first year engineering design as a foundation course for Virginia Tech's undergraduate engineering degree programs. She holds bachelors and masters degrees from Rutgers University, Lehigh University and Colorado School of Mines, and studies best practices in pedagogy, reflective learning and critical thinking as aids to enhanced student learning.

# You Might (or Might Not) Know More Than You Thought: Student Self-Perception vs. Performance in First Year Engineering Graphics and Computer Programming

Natalie Van Tyne
Virginia Polytechnic Institute, nvantyne@vt.edu

*Abstract* - **The results of a beginning of semester survey of students' current abilities in engineering graphics and computer programming in an introductory engineering design course were compared to their homework assignment and test grades in engineering graphics and computer programming. The graphics unit consisted of four weeks of manual drafting followed by three weeks of computer-aided drawing (CAD) with Autodesk Inventor. The programming unit, lasting six weeks, consisted of review and expansion of MatLab skills and tools. The course also included a semester-long design project, along with instruction and practice in technical communication and teamwork.**

**With respect to the graphics and programming knowledge area, the researcher categorized the participants as Beginner, Average or Experienced, depending on their descriptions of their exposure and perceptions of either subject. The distribution for initial graphics ability was approximately 30% Beginner, 40% Average and 30% Experienced. For programming, the distribution was approximately 35% Beginner, 45% Average and 20% Experienced.**

**A final reflective survey asked students to identify the most important areas of knowledge learned in the course, and why they were important to learn, as well as what areas were still difficult for them. Many students identified graphics and programming as among the most important areas for them to learn, as well as areas that they continued to find to be difficult.**

**Results revealed noticeable differences among the average grades, frequency of the most important area of knowledge and frequency of remaining difficulty between the two knowledge areas, with lower average grades and greater frequencies of importance and difficulty for programming than for graphics. These differences can be traced to participants' initial and final impressions of both knowledge areas in terms of certain elements of self-efficacy, namely performance accomplishment, physiological state, and extrinsic utility value.**

*Index Terms* – Self-efficacy, reflection, first-year design.

## INTRODUCTION

Students' perceptions of their abilities in fundamental engineering skills such as graphics and computer programming may be influenced by their familiarity with these skills, as well as their assessment of how well they were able to perform them upon exposure and practice. While some students may doubt their ability to master these skills, others possess a sufficient level of confidence and persistence to overcome any doubt about their current or future ability. The similarity between belief in one's ability to acquire a particular skill that is practiced in a specific context (self-efficacy), and the belief that one can be successful (self-confidence) may also lead some students to conclude that they can't become "good" at something if they can't be successful at it on the first or second attempt. This threatens their self-confidence and may affect self-efficacy. Lack of sufficient prior exposure to, or extent of experience with, a particular engineering tool, such as graphics or computer programming, may, or may not, lead to conclusions that particular students can, or cannot, become adept at using them. Other students, however, may be surprised to find that they can be more successful at acquiring and even mastering a technical skill than they originally imagined, where their initial impression belies their eventual achievement.

Because of the variations between students' initial perceptions of their ability to acquire skills in engineering graphics and computer programming, based on prior experience or the lack of it, this study is intended to answer this research question:

- How do first year engineering students' perceptions of their abilities to solve engineering graphics and computer programming problems align with their eventual proficiency as measured by homework grades and test grades?

## PROJECT BACKGROUND AND PURPOSE

Self-efficacy among engineering students has been studied widely, especially with respect to retention in engineering programs among under-represented groups, such as women. [1-6] It has been further related to retention through the application of expectancy-value theory with several constructs: intrinsic interest value, attainment value and extrinsic utility value. [1] These constructs apply to both men and women students, and embody a number of attitudes that first year students develop as a result of their current and prior exposure to various technical and non-technical aspects of engineering. Specific experiences that influence these attitudes may arise from individual exposures to technical knowledge areas that students often associate with engineering, such as graphics and computer programming.

Sources for these factors can be found from studies drawing on self-efficacy theory, as first identified by Bandura and applied often since that time. [7-9] Bandura identifies four major influences on self-efficacy: performance accomplishments, vicarious experiences, verbal persuasion and physiological states.[2] The first is the direct result of what a student does to practice a skill such as graphics or programming, and how much they are able to accomplish toward a solution to a homework or test problem in these knowledge areas. An example of the second influence, vicarious experience, is found in modeling, where an example problem is worked out and the student recognizes that they understand what was done to the extent that they could replicate it with another problem.[2] Verbal persuasion may be given by an instructor, exhorting the student to try a problem by showing the instructor's confidence in the student's ability to solve it, or by self-instruction, where the student successfully figures out how to solve the problem on their own. [2] Finally, the physiological state of emotional arousal can affect self-efficacy when a student feels a positive sense of accomplishment when a problem is solved, or dismay when they struggle with a problem until they give up on it due to a perceived sense of diminishing return. Since many engineering students consider their past experience as predictive of continued success or failure, performance accomplishments are expected to be the strongest influence on engineering students' self-efficacy.

## METHODOLOGY

Our institution is a large university in the eastern United States, where the introductory engineering course sequence is organized into sections of 30 students each. While the first course is concentrated in the formation of an engineering identity and plan of study, a selection of professional and technical skills, including an introduction to computer programming, the second course serves as an introduction to engineering design, with a semester-long design project, and instruction and practice in a wider variety of professional and technical skills, including engineering graphics and an expansion of computer programming knowledge. The design project requires to use of both graphics and programming in its solution. All graphics and programming instruction and practice were conducted using a problem-based format.

This study design is primarily quantitative, since it contains an initial survey about perceived abilities in graphics and programming, treatment in the form of instruction, practice and grading in these knowledge areas, and a final reflective survey. Results from these surveys enabled us to identify aspects of self-efficacy that may have influenced final proficiency in graphics or programming.

*Study Context*

This study involved the second of two one-semester introductory engineering courses, in which students pursued a design project in teams for 15 weeks, coupled with individual development in engineering graphics and computer programming. On the first day of the course, students were asked to provide feedback, through a numeric scale and written comments, about their current abilities in graphics and programming. A total of 10 graphics-related and 7 programming-related combinations of homework and tests were then completed, sequentially within each knowledge area, over Weeks 1-13 of the course, with 7 weeks devoted to graphics and 6 weeks to programming in MatLab. The reflective survey asked students to identify the most important areas of knowledge learned in the course, and why they were important to learn, as well as what areas were still difficult for them. Many students identified graphics and programming as among the most important areas for them to learn, as well as areas that they continued to find to be difficult.

*Participants*

The participants were first year engineering students in three sections of the second semester introductory engineering design course, who had completed the initial and final surveys and course assignments and tests described above. Relatively few of the participants had prior experience with engineering graphics, while the majority had some experience with MatLab and/or other programming languages.

*Data Collection*

Student-derived data were collected and de-identified in accordance with Institutional Review Board policy. For each knowledge area, participants were grouped into three categories: Beginner, Average and Experienced, depending on their previous experience in these skill areas and expressed attitudes toward them. These categories are described as follows:

**Table 1: Students' Perceived Ability in Graphics and Programming**

| *Perceived Ability* | *Graphics* | *Programming* |
|---|---|---|
| Beginner | No prior experience with sketching or CAD | No prior experience, low comfort level, and/or dislikes it |
| Average | Some exposure to sketching and/or CAD; may have forgotten it | Moderate comfort level with one or more programming languages |
| Experienced | One or more semesters of previous CAD experience and/or CAD certification | High comfort level with MatLab and one or more other programming languages |

The distribution for initial graphics ability was approximately 30% Beginner, 40% Average and 30% Experienced. For programming, the distribution was approximately 35% Beginner, 45% Average and 20% Experienced.

Homework grades and test grades were combined into overall percentages, on a 100% scale for each knowledge area, respectively. The percentages were then ordered from highest to lowest within each perceived ability category.

Finally, the reflective survey contained the following questions, for which a number of students mentioned graphics and/or programming:

- What were the most important areas of knowledge that you learned in this course?
- What did you learn in our course that you still find difficult?

*Data Analysis*

Graphics-based and programming-based data were analyzed separately, since it was our intent to reveal aspects of self-efficacy that might differ between these two knowledge areas. This assumption was informed by a greater amount of student feedback about programming than about graphics, and specifically about MatLab, at the end of the course. The average percentage grade was calculated in all three perceived ability categories for both knowledge areas, and the student responses were scrutinized for indications of any applicable Bandura factors of self-efficacy, i.e., performance accomplishments, vicarious experiences, verbal persuasion or physiological states. [2]

The numbers of responses identifying graphics or programming as one of the most important knowledge areas in the course were tabulated, along with responses indicating difficulty in either knowledge area. Specific skills that were difficult were noted for additional reinforcement or emphasis in future semesters.

## RESULTS

When each student's survey results were compared to their combined homework and test percentage grades in graphics and in programming, we found that students with prior experience usually earned higher grades on graphics homework and tests than those with no experience. However, prior experience with MatLab did not guarantee success in our programming unit, and prior experience with Java, C++ or Python, without MatLab, yielded mixed results at best.

Results for percentage grades among the three ability categories, as well as the percentages of students in each category who identified graphics or programming as most important and/or still difficult, are summarized in Tables 2 and 3 below:

**Table 2: Final Perceptions and Grades for Graphics Among Three Initial Perception Categories**

| *Description* | *Beginner* | *Average* | *Experienced* |
|---|---|---|---|
| Number of participants | 19 | 23 | 15 |
| Average % grade for all assignments and tests | 86.8 | 83.4 | 91.1 |
| % of participants for whom graphics skills were most important in the course | 89.5 | 39.1 | 33.3 |
| % of participants for whom graphics is still difficult | 31.6 | 13.0 | 6.7 |

**Table 3: Final Perceptions and Grades for Programming Among Three Initial Perception Categories**

| *Description* | *Beginner* | *Average* | *Experienced* |
|---|---|---|---|
| Number of participants | 19 | 26 | 12 |
| Average % grade for all assignments and tests | 73.2 | 80.1 | 82.8 |
| % of participants for whom programming skills were most important in the course | 42.1 | 50.0 | 25.0 |
| % of participants for whom programming is still difficult | 68.4 | 30.8 | 25.0 |

On an overall basis, only Average participants considered programming to be a Most Important knowledge area more often than for graphics, although a greater percentage of participants in all three categories considered programming to still be difficult than those for whom graphics was still difficult.

Participants classified as Beginners in graphics stated that they had little to no exposure in any form of graphics (i.e., manual sketching or CAD), and therefore were not expected to have any positive or negative impressions of it. These participants were also much more likely to identify graphics as one of the most important knowledge areas in the course. Conversely, programming Beginners often expressed negative impressions, in that they were uncomfortable with or loathsome toward whatever programming language they had encountered.

Participants classified as Average were similarly open-minded about graphics, based on little to no opportunity to form a negative impression, although their final average percentage grade was lower than that for Beginners. Participants who were Average in programming expressed a moderate comfort level due to prior experience with MatLab and/or other programming languages that was positive in nature.

The Experienced participants in graphics had substantial experience with CAD programs compared to those in the other two categories, and were less likely to identify graphics as a Most Important knowledge area than other participants. Their average percentage grade was higher than those in the other two categories for both graphics and programming, although those Experienced in programming achieved considerably lower overall percentage grades than those Experienced in graphics.

Since the final reflective survey was intended to invite consideration of all of the acquired knowledge of the course, not just graphics or programming, approximately 40% of the participants did not mention graphics as among their most important items of knowledge. Similarly, 30% of the participants did not mention programming (i.e., MatLab). We were not able to determine if there was a correlation between any student's intended engineering major and their identification of either technical area as a "most important" response.

This study is further limited in that a complete suite of descriptive statistics was not employed to further describe the distribution of graphics and programming assignments and test grades. Instead, we sought to identify an overall trend toward low or high grades among the Beginner, Average and Experienced cohorts.

## DISCUSSION

The participants in the Beginner, Average and Experienced categories are more or less distributed as we would expect for both knowledge areas, based on our criteria for assigning participants to their respective categories. It is also logical to expect students to be more highly skilled in certain technical areas than others, based on aptitude, experience and preference, and there could be multiple reasons why a participant with high proficiency in graphics may not be equally skilled in programming. Students could also easily justify the importance of mastering graphics within the context of a design project that required its use. They could also recognize this area of knowledge as being useful to their career in the long term, because they identify graphics with engineering. This is an example of extrinsic utility as a value-related construct for self-efficacy.[1] A similar identification was not found for programming, mainly because fewer participants considered programming to be among the most important knowledge areas in the course.

However, we believe that there are certain Bandura-based aspects of self-efficacy that could explain the noticeable difference in average percentage grades between graphics and programming, as well as the higher percentages of students who still found programming to be difficult by the end of the course, compared to those who still found graphics to be difficult. [2] The two most notable self-efficacy factors found in participant responses about programming were performance accomplishments and physiological states.[2] Beginner participants were more likely than either Average or Experienced participants to be uncomfortable with programming or even to dislike it (emotion as a physiological state), because they had concluded that they could not be successful at it (performance accomplishment). By contrast, Average participants might not necessarily enjoy programming, but had likely experienced some level of success with it, and therefore did not dislike it (performance accomplishment). While vicarious experiences and verbal persuasion might have had an effect on some participants in either of these categories, it is doubtful that either factor would have influenced participant self-efficacy toward programming.

## CONCLUSIONS AND RECOMMENDATIONS

Further application of Bandura's theory by Chemers, Hu and Garcia reveals a relationship between high self-efficacy and lack of stress or anxiety about completing tasks, to the point of readily using metacognitive strategies such as planning and self-regulation.[3] Goal-setting is also an important aspect of planning and self-regulation, as goals provide a means to measure what has been accomplished, which contributes to performance accomplishment.[8-9] A common goal for students is to perform well enough in a knowledge area to receive an A grade on a test, while a longer-term goal could be to understand the concepts behind specific compositional or computational steps in order to apply them to an ill-defined problem that does not have just one "right" answer. First-year students may already have an idea about what they want to achieve, but not necessarily how to achieve it.

Additional insight may be gained from a more detailed analysis of assignment and test grades as functions of perceived difficulty and importance in the course. This may, or may not, reveal to what extent a student's perception of

their abilities in graphics and/or programming would affect their performance in these areas, or if there is a correlation between performance and importance.

It is up to instructors to employ instructional methods emphasizing performance accomplishment, especially in programming, in part by encouraging whatever progress can be made. An improvement in students' self-efficacy with programming is a potential example of persistence and tangible accomplishment leading to the acceptance of more complex and less-well-defined challenges such as open-ended problem solving. While programming-based careers seem to be appealing in the current economy, and information management grows in importance, these are not enough to overcome lingering student disappointment with their lack of early success in programming proficiency. In fact, these factors may also contribute to a general lack of self-efficacy among our students with proficiency in computer programming, which appears to be of less concern than the lack of self-efficacy with proficiency in engineering graphics.

## REFERENCES


[1] B.D.Jones, M.C. Paretti, S.F. Hein, T.W. Knott, "An Analysis of Motivation Constructs with First-Year Engineering Students: Relationships Among Expectancies, Values, Achievement and Career Plans," Journal of Engineering Education, Vol. 99, pp. 319-336, 2010.

[2] A. Bandura, "Self-Efficacy: Toward a Unifying Theory of Behavioral Change," Psychological Review, vol. 84, No. 2, pp. 191-215, 1977.

[3] M.M. Chemers, L.Hu, B.F. Garcia, "Academic Self-Efficacy and First-Year College Student Performance and Adjustment," Journal of Educational Psychology, Vol. 93, No. 1, pp. 55-64, 2001.

[4] R. Marra, K. Rodgers, D. Shen, and B. Bogue, "Women Engineering Students and Self-Efficacy: A Multi-Year, Multi-Institution Study of Women Engineering Student Self-Efficacy," Journal of Engineering Education, vol. 98, no. 1, pp. 27-38, 2009.

[5] R. Felder, G. Felder, M. Mauney, C. Hamrin, and E. Dietz,, "A Longitudinal Study of Engineering Student Performance and Retention. III. Gender Differences in Student Performance and Attitudes," Journal of Engineering Education, vol. 84, pp. 151-163, 1995.

[6] G. Sonnert, Who Succeeds in Science?, New Brunswick, NJ, USA: Rutgers University Press, 1995, pp. 1-15, 138-157.

[7] Nye, "Strong Objectivity and the Language of Science," in Women Succeeding in the Sciences =-Theory and Practice Across Disciplines, West Lafayette, IN: Purdue University Press, 2000, pp. 195-203.

[8] S. Rosser, "Conclusion: Changing Curriculum and Pedagogy to Reach the Majority Results in a Positive Upward Spiral," in Teaching the Majority: Breaking the Gender Barrier in Science, Mathematics and Engineering, New York, NY: Teachers College Press, 1995, pp. 220-229.

[9] N. Van Tyne and M. Brunhart-Lupo, "Ethics for the "Me" Generation: How "Millennial" Engineering Students View Ethical Responsiibility in the Engineering Profession," in *American Society for Engineering Education Annual Conference and Exposition 2015*, Seattle, WA, 2015.


## AUTHOR INFORMATION


Natalie Van Tyne, Associate Professor of Practice, Department of Engineering Education, Virginia Polytechnic Institute and State University, nvantyne@vt.edu.