The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #42772

Differences in Attitudes and Self-efficacy Toward Programming of Students in Mechanical and Industrial Engineering Programs

Xinyi Ma, University of Toronto

Xinyi Ma is a graduate student researcher at University of Toronto in the Department of Mechanical and Industrial Engineering, supervised by Prof. Janet Lam. Her research interest is student experience in engineering education. Xinyi holds an Honours Bachelor of Science in Statistics with a minor in Computer Science from University of Toronto.

Janet Lam, University of Toronto

Janet Lam is an Assistant Professor, Teaching Stream in operations research with the Department of Mechanical and Industrial Engineering.

She has been working in the field of maintenance optimization since 2008, with an emphasis on optimal scheduling of inspections for condition-based maintenance. Janet served as a research associate at the Centre for Maintenance Optimization and Reliability Engineering (C-MORE) applying academic research directly with industry partners, including those in mining, utilities, transportation, and the military. Janet has a track record of cultivating strong relationships with industry partners and developing maintenance engineering resources that are both useful and current.

She is also a respected engineering educator with more than 10 years of teaching undergraduate, graduate, and professional students. She was a Teaching Specialist for first year engineering students at Michigan State University from 2016 to 2017. She is a Fellow of the National Effective Teaching Institute and a Runner-Up Best Upper Year Instructor in the Skule Student Choice awards 2020-2021.

Janet received her Ph.D. in Industrial Engineering at the University of Toronto.

Work in Progress: Differences in Attitudes and Self-efficacy toward Programming of Students in Mechanical and Industrial Engineering Programs

This WIP research stems from the significance of computer programming nowadays and explores the subjective perspectives and experiences about programming. In the context of a third-year quality control course with a computing lab component involving both mechanical and industrial engineering students, we aim to answer the following research questions: (1) How do differences in the programs' curricula impact students' attitudes and self-efficacy toward programming? (2) How do the computing lab activities affect the students' attitudes and self-efficacy toward programming? Three surveys are distributed across the semester, collecting students' programming backgrounds, perceived interests, usefulness, and self-efficacy, along with their engagement with lab activities. Initial findings indicate that industrial engineering students have more positive attitudes and higher confidence toward programming compared to mechanical engineering students. Future research will further investigate this question with the following survey responses and seek to understand the influence of programming lab activities on students' programming experiences.

Keywords: programming, attitudes, self-efficacy, mechanical engineering, industrial engineering

Introduction

As computer programming has been widely used in both academic research and industrial practice, the skill is becoming increasingly important in engineering education. According to A. Bandura, self-efficacy accurately predicts both subsequent behaviors and outcomes [1], and self-efficacy toward programming could reflect confidence in performing tasks such as understanding the logical structures, solving problems through programming, and debugging [2], [3]. Meanwhile, attitudes also play an important role in students' learning outcomes, and computer programming attitudes could include different dimensions like programming interest and mindset in various relevant scales [4], [5].

Several studies have investigated students' attitudes and self-efficacy towards programming, concluding a positive relationship between programming attitudes and self-efficacy, with explorations in gender, learning style, and programming experience factors in attitudes and self-efficacy, while they focused on students in computer science and engineering majors [6] - [8]. Looking at studies conducted in courses with programming components intended for general engineering students, Ronan and Erdil researched whether a first-year rotation-based survey course introducing various computer science and engineering topics would impact attitudes and beliefs about computing of students but found no significant differences [9]. Meanwhile, a second-year course including three computational projects using scaffolding pedagogy led to positive changes in programming self-efficacy for biomedical and agricultural engineering undergraduates, while the results in other aspects like interest, anxiety, and mindset are mixed [10]. In another study, a significant positive change in attitudes in an introductory programming course for students from non-programming majors was observed, as they encouraged students to work on coding scenarios in peer group activities [11]. Also, as students from several engineering streams excluding software and electrical engineering were surveyed, it was found perceived usefulness of programming plays an important role in students' intention to take programming classes in college [12].

While these studies provide important insights into programming attitudes and self-efficacy, they do not address the potential differences between students from various engineering streams created by the different curricula. There have been proposals to integrate computational components through successive engineering curricula, strengthening the students' programming skills applicable to specific disciplines [13]. Shiavi and Brodersen also explored students' preferences for instructional modes in an introductory computing course and discovered they prefer laboratory over lecture and consider laboratory settings useful especially when learning more challenging themes [14]. In this project, we will compare the attitudes and self-efficacy toward programming of mechanical and industrial engineering students and analyze how the curricula of the two streams prepare students for programming mentally and emotionally. The research questions are: (1) How do differences in mechanical and industrial engineering curricula impact students' attitudes and self-efficacy toward programming? (2) How do the computing lab activities affect the students' attitudes and self-efficacy toward programming?

Methodology

Context and Participants

The study is conducted in a third-year quality control course at the University of Toronto. This course is a required core course for industrial engineering students, and a technical elective under the manufacturing stream for mechanical engineering students, while it could also be taken by students in other engineering streams as an elective. There are 114 industrial engineering students and a total of 58 mechanical and other engineering students enrolled in the course in the winter 2024 term. The course components for industrial and mechanical engineering students are different: while both groups have weekly 3-hour lectures, mechanical students have 3-hour labs, and industrial students have 1-hour labs and 2-hour tutorials. The contents covered in those sections are the same, but students have access to the school computers during computing labs while they have to bring their own laptops for tutorials as these sessions are in regular classrooms. However, since most students have their laptops and use them in the practical sessions, the difference in settings might not be the main influence factor. The computing labs, which are referred to as Tutorilabs in this course, cover basic data analytics skills using Python, with an emphasis on data manipulation and visualization to solve practical problems using common packages such as NumPy, Pandas, and Matplotlib.

Students in both majors have taken the same required courses in their first year, including a course introducing computer programming fundamentals. The industrial engineering curriculum includes core courses on object-oriented programming, data science, and data structures and algorithms, while there are no such requirements in the mechanical engineering curriculum. Therefore, we suppose industrial engineering students generally have more programming experience than mechanical engineering students prior to this course, though more previous exposure does not necessarily lead to more positive attitudes and higher self-efficacy toward programming.

Data Collection

Data are collected through three surveys distributed in the first week of the course, the week after midterm in early March when students have attended 5 out of 10 labs, and at the end of the semester by April 2024. The first survey asks about students' programming background, which they select from a list of programming languages and environments they have been exposed to. Then they rate their interest in programming, their perceived usefulness of programming, and their self-confidence about programming on a 5-point Likert scale. In the midterm and final surveys, we also include questions about students' engagement in lab activities, asking how the lab activities help them in the course assessments, how the courses they have taken prepare them for the programming tasks, and their perceived performance for the course besides questions about attitudes and self-efficacy, to further understand the role of previous and current courses in students' development of mental and emotional view toward computer programming.

Questions in the first survey are as follows:

1. Please indicate your previous exposure to programming (select one or more items you have experience with from the list):
    - Python
    - R
    - Java
    - MATLAB
    - C
    - JupyterHub
    - GitHub
    - Other (Text input)
2. How would you rate your interest towards programming?
   1 – Not At All Interested 2 – Not Interested 3 – Neutral 4 – Interested
   5 – Extremely Interested
3. Do you think programming is useful for your future studies/career?
   1 – Not At All Useful 2 – Not Useful 3 – Neutral 4 – Useful 5 – Very Useful
4. Comparing to your peers, how would you rate your programming abilities?
   1 – Significantly Below Average 2 – Below Average 3 – Average 4 – Above Average
   5 – Significantly Above Average

Questions in the mid-term and end-of-course surveys are as follows:

1. How much have you attended the Tutorilab sessions or made use of the Tutorilab files?
   1 – Not At All (skip Q2 if this option is selected) 2 – A little 3 – Some 4 – Much
   5 – Very Much
2. How much have the Tutorilab activities prepare you for relevant assessments for this course?
   1 – Not At All 2 – A little 3 – Some 4 – Much 5 – Very Much
3. How much have the previous courses you have taken prepared you for the programming components of assessments?
   1 – Not At All 2 – A little 3 – Some 4 – Much 5 – Very Much
4. Which course have you taken best prepared you for the programming components of assessments? (Text input for the course code)

5. How would you rate your interest towards programming?
   1 – Not At All Interested 2 – Not Interested 3 – Neutral 4 – Interested
   5 – Extremely Interested
6. Do you think programming is useful for your future studies/career?
   1 – Not At All Useful 2 – Not Useful 3 – Neutral 4 – Useful 5 – Very Useful
7. Comparing to your peers, how would you rate your programming abilities?
   1 – Significantly Below Average 2 – Below Average 3 – Average 4 – Above Average
   5 – Significantly Above Average
8. What is your expected score for this course?
   90-100 80-89 70-79 60-69 50-59 0-49

As our research focuses on the cognitive perspective rather than programming ability, we do not collect students' real grades for this course. Instead, we ask for their expected score for this course at mid and end of the term and match this perceived performance with their other responses. The data collection methodology has been approved by the Research Ethics Board (REB) at the University of Toronto.

The first survey has a total of N = 83 responses, where 47 are industrial engineering students, 23 are mechanical engineering students, and there are also 12 materials science engineering students and 1 chemical engineering student included in the responses. Due to the smaller sample sizes of materials science and chemical engineering student groups, only industrial and mechanical engineering groups are included in the statistical analysis, while the exploratory data analysis includes the distribution of students from all 4 streams to show some side findings in addition to the main research questions.

Analysis

Welch's t-test is used to compare differences in the perceived mean levels of programming interest, usefulness, and efficacy between mechanical and industrial engineering students, which is more robust to sample size and variance differences between the two groups compared to the Student's t-test [15]. While nonparametric tests such as the Mann-Whitney U test could be used for ordinal data from Likert scales, it is still suggested to apply parametric tests if both groups have sample sizes larger than n=15 even when some test assumptions are not met [16].

When data collection from the mid-term and end-of-course surveys are completed, we propose to use two-way mixed ANOVA to measure how the two groups of students' programming attitudes and self-efficacy evolve over the semester. Ordinal logistic regression might also be conducted to take more factors that could affect attitudes and efficacy levels into account. Besides, qualitative analysis will also be performed on the courses they have taken and the courses they think that have prepared them for the lab activities to provide additional information on the findings.

Results

According to the survey data, previous exposure to Python and MATLAB is rather high in all students, while exposure to other programming languages or environments (R, Java, C, JupyterHub, GitHub) is much higher among industrial engineering students than other engineering students. The results are expected as the industrial engineering program requires more programming-related courses than the other three programs, providing them with opportunities to work with a wider range of programming tools. While this course mainly uses Python, the background differences could still affect students' general programming attitudes.

Industrial engineering students are generally interested in programming, and mechanical and materials science students are closer to the neutral level of interest. Students in all groups acknowledge that programming is useful for their future studies and careers, while the mean and median levels are still higher among industrial engineering students than students from other streams. The median self-efficacy ratings for these groups of students are all around average, but the mean levels indicate more confidence in industrial engineering students than in the two groups. Variance differences across groups are not huge and are not an influencing factor in Welch's t-test.

Table 1.
Descriptive statistics of perceived programming interest, usefulness, and self-efficacy for industrial, mechanical, materials science, and chemical engineering students

Items	Interest			Usefulness			Self-Efficacy		
Programs	Median	Mean	Var	Median	Mean	Var	Median	Mean	Var
Industrial (n=47)	4.00	3.96	0.93	5.00	4.62	0.36	3.00	3.32	0.73
Mechanical (n=23)	3.00	3.30	0.65	4.00	3.91	0.95	3.00	2.74	0.54
Materials Science (n=12)	3.50	3.08	1.24	4.00	4.00	0.33	3.00	2.58	0.58
Chemical (n=1)	4.00	4.00	0.00	4.00	4.00	0.00	2.00	2.00	0.00

Note: There is only 1 student response from chemical engineering, so the above statistics for the group only reflect the answer of this student.

Then, we apply Welch's t-test with the hypotheses H_0: The mean levels of programming interest, usefulness, and self-efficacy of mechanical and industrial engineering students are the same respectively, and H_A: The mean levels of programming interest, usefulness, and self-efficacy of mechanical and industrial engineering students are different.

As p-values for each of the Welch's t-tests for interest, usefulness, and self-efficacy are smaller than the significance level .05 (and are all below .01), we would reject the null hypotheses. Results have indicated statistically significant differences between mean levels of programming interest, usefulness, and self-efficacy between the two groups of mechanical and industrial engineering students.

Table 2.
Welch's t-test results summary table for mean levels of programming interest, usefulness, and self-efficacy of students in mechanical and industrial engineering programs

Items	Interest**			Usefulness**			Self-Efficacy**		
Values	Test Statistic	P-value	Df	Test Statistic	P-value	Df	Test Statistic	P-value	Df
	2.930	0.005	51.220	3.116	0.004	30.323	2.886	0.006	49.616

Note: *P < 0.05, **P < 0.01.

Discussion

The initial findings have provided some insights into our first research question, as there are differences in mechanical and industrial engineering students' attitudes and self-efficacy toward programming, where industrial students perceive higher interest, usefulness, and confidence toward programming compared to mechanical students, which could be influenced by the programs' curricula. We will dive deeper into this question with data from the following two surveys. Meanwhile, with responses about their engagements in lab activities, we seek to learn the effect of computing lab on their programming affection, motivation, and perceived confidence. In the context of this specific course with connections to their previous courses, we hope to gain more understanding of engineering curricula' influence on students' perspective of computer programming.

These results would first be useful for engineering educators at the University of Toronto. Mechanical and industrial engineering have been two engineering streams under the same department, and while the two programs have very different curricula from the second year, some instructors in the upper years do not fully recognize variations between these two groups. With more knowledge of the student population, educators could provide help to students more accordingly when the two groups are mixed in a course. The course studied will be discontinued as a required course for industrial engineering students starting from next year, and the two sections will be combined with students from both streams taking it as a technical elective. In such kind of case, it will be difficult for the instructors and teaching assistants to efficiently lead mixed sessions with students of various backgrounds and skill levels, as pace differences between the industrial and mechanical sections are already starting to show during the first weeks of the course. We hope our research throughout the semester can assist in the transit and provide information for similar engineering courses with a diverse student body.

From a more general perspective, while curricula of different engineering streams need to focus on the specific field and it is not realistic to require all engineering students to take numerous programming-related courses, incorporating programming components applicable to the subject would be beneficial, as Raubenheimer et al has mentioned in their work [13]. Tasks like data analyzing and visualizing are inevitable in most engineering fields, and the relevant preparations in undergraduate courses not only expose them to programming skills but might also help them build up interest and confidence.

Finally, while we encourage participation through various methods, we could not get responses from all students registered in the course, and we expect the final sample size of three matched data sets to be lower than the N = 83 we have now. This would be a limitation for our project, and we hope future work would be conducted in a larger quantity and also include students from more engineering streams.

References

[1] A. Bandura, "Self-efficacy: toward a unifying theory of behavioral change," *Psychological Review*, vol. 84, no. 2, pp. 191–215, 1977.

[2] M.-J. Tsai, C.-Y. Wang, and P.-F. Hsu, "Developing the Computer Programming Self-Efficacy Scale for computer literacy education," *Journal of Educational Computing Research*, vol. 56, no. 8, pp. 1345–1360, 2019.

[3] M. J. Scott and G. Ghinea, "Measuring enrichment: The assembly and validation of an instrument to assess student self-beliefs in CS1," in *Proceedings of the tenth annual conference on International computing education research*, 2014.

[4] I. Cetin and M. Y. Ozden, "Development of computer programming attitude scale for university students," *Computer Applications in Engineering Education*, vol. 23, no. 5, pp. 667–672, 2015.

[5] B. Dorn and A. Elliott Tew, "Empirical validation and application of the computing attitudes survey," *Computer Science Education*, vol. 25, no. 1, pp. 1–36, 2015.

[6] O. Ozyurt, "An analysis on distance education computer programming students' attitudes regarding programming and their self-efficacy for programming," *Turkish Online Journal of Distance Education*, vol. 16, no. 2, 2015.

[7] M. S. Gunbatar, "Examination of undergraduate and associate degree students' computer programming attitude and self-efficacy according to thinking style, gender and experience," *Contemporary Educational Technology*, vol. 9, no. 4, pp. 354–373, 2018.

[8] M. Derya GURER, I. Cetin, and E. Top, "Factors affecting students' attitudes toward computer programming," *Informatics in Education*, vol. 18, no. 2, pp. 281–296, 2019.

[9] D. Ronan and D. Cenk Erdil, "Impact on computing attitudes and career intentions in a rotation-based survey course," in *2020 ASEE Virtual Annual Conference Content Access Proceedings*, Jun. 2020.

[10] J. Arigye, A. J. Magana, J. A. Lyon, and E. Pienaar, "Biomedical and agricultural engineering undergraduate students programming self-beliefs and changes resulting from computational pedagogy," in *2023 ASEE Annual Conference & Exposition Proceedings*, Jun. 2023.

[11] D. Whittinghill, D. Nelson, K. A. R. Richards, and C. Calahan, "Improving the affective element in introductory programming coursework for the 'non programmer' student," in *2014 ASEE Annual Conference & Exposition Proceedings*, Jun. 2014.

[12] K. Steelman et al., "Work in progress: Student perception of computer programming within engineering education: An investigation of attitudes, beliefs, and behaviors," in *2020 ASEE Virtual Annual Conference Content Access Proceedings*, Jun. 2020.

[13] D. Raubenheimer, R. Brent, J. Joines, and A. Craig, "Integration of computer-based problem solving into engineering curricula," in *2008 ASEE Annual Conference & Exposition Proceedings*, Jun. 2008.

[14] R. Shiavi and A. Brodersen, "Study of instructional modes for introductory computing," *Journal of Engineering Education*, vol. 94, no. 4, pp. 355–362, 2005.

[15] M. Delacre, D. Lakens, and C. Leys, "Why psychologists should by default use Welch's t-test instead of Student's t-test," *International Review of Social Psychology*, vol. 30, no. 1, pp. 92–101, 2017.

[16] G. M. Sullivan and A. R. Artino Jr, “Analyzing and interpreting data from likert-type scales,” *Journal of Graduate Medical Education*, vol. 5, no. 4, pp. 541–542, 2013.