First-Year Engineering Student Perceptions in Programming Self-Efficacy and the Effectiveness of Associated Pedagogy Delivered via an Introductory, Two-Course Sequence in Engineering

Dr. James E. Lewis, University of Louisville

James E. Lewis, Ph.D. is an Assistant Professor in the Department of Engineering Fundamentals in the J. B. Speed School of Engineering at the University of Louisville. His research interests include parallel and distributed computer systems, cryptography, engineering education, undergraduate retention and technology (Tablet PCs) used in the classroom.

Dr. Brian Scott Robinson, University of Louisville
Mr. Nicholas Hawkins, University of Louisville

Nicholas Hawkins is a Graduate Teaching Assistance in the Engineering Fundamentals Department at the University of Louisville. A PhD student in Electrical and Computer Engineering, he received both his B.S. and M. Eng. from the University of Louisville in the same field. His research interests include power electronics and controls, as well as engineering education for first-year students.

First-Year Engineering Student Perceptions in Programming Self-Efficacy and the Effectiveness of Associated Pedagogy Delivered via an Introductory, Two-Course Sequence in Engineering

In the fall of 2014, the J. B. Speed School of Engineering (SSoE) at the University of Louisville (UofL) commenced an endeavor to renovate the school's existing course(s) focused on introducing first-year students to the profession and fundamentals of engineering, resulting in a two-course sequence that all first-year SSoE students are required to take. The first component of this sequence, *Engineering Methods, Tools, & Practice I* (ENGR 110), is structurally analogous to the previously existing introductory course and is primarily focused on introduction to and practice with fundamental engineering skills. The second component, *Engineering Methods, Tools, and Practice II* (ENGR 111), is a makerspace-based course primarily focused on application and integration of the fundamentals learned in ENGR 110. Included amongst numerous skills institutionally-identified as "fundamental" was programming, hence all SSoE engineering students – regardless of discipline – are exposed to edification in the basics of programming.

Associated programming curriculum developed for this sequence was heavily influenced by a desire to reflect the varying nature of programming applications throughout industry and the engineering profession. In other words, it is virtually impossible to expose students to all of the possible programming "styles" and dozens of varying programming languages rampant in the modern work force. Accordingly, pedagogy throughout both ENGR 110 and 111 has been designed to expose students to multiple types of programming interfaces with a key focus on understanding programming fundamentals that remain essentially unchanged regardless of the methodology and/or language.

Programming instruction in ENGR 110 involves a five-week introduction to fundamental programming concepts through the Python language. This instruction utilizes the zyBooks online educational platform as an interactive e-text, while simultaneously employing team-based instruction through in-class collaborative activities. The ENGR 110 curriculum culminates in a comprehensive Vector Project, which, while involving other skills developed throughout the course, also requires students to develop useful programs in Python from realistic constraints.

ENGR 111 culminates in team-based Cornerstone projects that all students demonstrate and present at the end of the semester. Throughout the semester up to Cornerstone demonstrations, course instruction, activities, and deliverables have been designed in a dual-purpose manner, in that they augment student practice of essential engineering skills (such as introductory programming), while at the same time scaffolding progression towards Cornerstone Project completion. Scaffolded lesson plans related to programming have been designed to expose students to two primary means of programming interface and methodology. These respectively include 1) Arduino-based platforms focused on instruction of algorithm-based programming methodology, and 2) Programmable Logic Controllers (PLCs) focused on instruction in ladder-logic based programming methodology. The Cornerstone Project for current course iteration(s) involves the construction and design of a windmill system; which includes the integration of a windmill, student-built AC motors, DC motors, circuitry, and data acquisition systems. Included within the Cornerstone demonstration assessment is a component dedicated to student-

programmed windmill parameter display. By means of integrated circuitry and programming executed via both the Arduinos and PLCs, Cornerstone demonstration(s) related to the programming aspect involves the inclusion of an LCD screen that displays five different, real-time windmill system parameters upon toggling of a pushbutton. The displayed parameters are (1) windmill speed (listed in revolutions per minute), (2) windmill system power output, (3) windmill blade efficiency, (4) windmill motor efficiency, and (5) windmill system efficiency.

In the Spring 2019 iteration of ENGR 111, students were provided with several quantitative and qualitative survey questions on their programming experience throughout the sequence, including perceptions related to comparison and preference of types, confidence level in basic programming, and the perceived usefulness of ENGR 110 curriculum in preparation for the ENGR 111 programming experience. Approximately two-thirds of the 443 students surveyed expressed preference of Arduino-based programming over PLC-based, while more than half of the students expressed appreciation that they were exposed to both interfaces. Expressed levels of confidence were evenly distributed across the spectrum ("extremely confident" to "not confident at all"), and the majority of surveyed students suggested that ENGR 110 pedagogy could be improved to further prepare students for programming in ENGR 111.

Introduction

In the fall of 2014, the J. B. Speed School of Engineering (SSoE) at the University of Louisville (UofL) commenced renovation to the school's existing course focused on introducing first-year students to the profession and fundamentals of engineering. Prior to the 2016 Fall semester, the incoming first-year students were required to take an introductory course primarily focused on introduction to and practice with fundamental engineering skills. Starting with the Fall semester in 2016, the required introductory course became two courses. The first component of this sequence, *Engineering Methods, Tools, & Practice I* (ENGR 110), is structurally analogous to the previously existing introductory course and is primarily focused on introduction to and practice with fundamental engineering skills. The second component, *Engineering Methods, Tools, and Practice II* (ENGR 111), is a makerspace-based course primarily focused on application and integration of the fundamentals learned in ENGR 110. One of the skills identified as fundamental to all majors was basic programming, so the fundamentals of programming are introduced in the ENGR 110 course. ENGR 111 requires the students to use their knowledge of programming in their end-of-semester Cornerstone Project.

The course sequence desires to expose the students to the varying nature of programming applications throughout the engineering profession. However, it is impossible to expose students to all possible programming styles and languages that are currently being used or developed in the modern workforce. The decision was made to expose the students to multiple types of programming in the ENGR 110/ENGR 111 course sequence. This decision was made since underlying programming fundamentals remain unchanged regardless of different language syntax, i.e. a for loop works the same even if the syntax is different.

The first introduction to programming in the sequence occurs in the ENGR 110 course. The programming instruction in this course is a five-week introduction to fundamental programming concepts using the programming language Python. The programming instruction uses zyBooks,

an online education platform with an interactive e-text, as well as in-class team-based instruction and collaborative activities. Some first-year students have never been exposed to formal programming, while some have previous programming experience.

The ENGR 111 course finishes with a team-based Cornerstone project that all students demonstrate and present at the end of the semester. The ENGR 111 course instruction, activities, and deliverables are designed to progress the students towards completion of their Cornerstone project. ENGR 111 is taught in a 15,000 ft^2 makerspace. The makerspace has an individual classroom, as well as a workstation area used as a laboratory workspace. This course allows for the Cornerstone project to be a long term project spanning multiple lab sessions, which is advantageous because it simulates how engineering functions in industry [1]. This involves hands-on learning and utilizes an instructor being present to assist the students [2] [3].

The current Cornerstone project involves the construction, and design of a windmill system. This system requires the integration of a windmill, student-built AC motors, DC motors, circuitry, data acquisition, manipulation of the acquired data, and the display of the data results.

The scaffolded lesson plans related to programming in ENGR 111 have been designed to expose the students to two different programming interfaces. The programming interfaces currently used in the Cornerstone project are Arduino based programing and programmable logic controllers (PLCs). By means of integrating circuitry and programming, the students use both the Arduino and PLC to gather data and display five different, real-time windmill system parameters to a LCD screen.

Arduino and PLC Combined Cornerstone Project Description

As mentioned in the Introduction, data acquisition for the Cornerstone project is driven by using an Arduino and a PLC. The Arduino Uno microcontroller is illustrated below in Figure 1 [4].

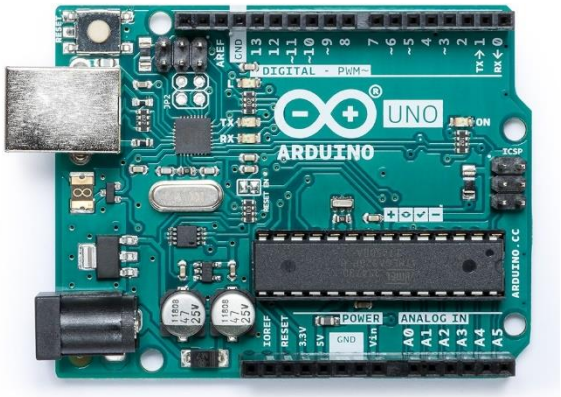

Figure 1: Arduino Uno Microcontroller

The Arduino Uno was chosen because it is excellent for teaching basic circuitry and programming. The Arduino Uno has easily accessible digital and analog input/output ports and uses a variant of the C programming language.

The PLC used in the Cornerstone project is an Allen Bradley MicroLogix 830 Programmable Logic Controller that is illustrated in Figure 2 [5]. Like the Arduino, the PLC includes several input and output ports, but these ports are only digital ports. The software used to program the PLC is Rockwell Automation's Connected Components Workbench (CCW) [6], a PLC ladder logic programming platform.

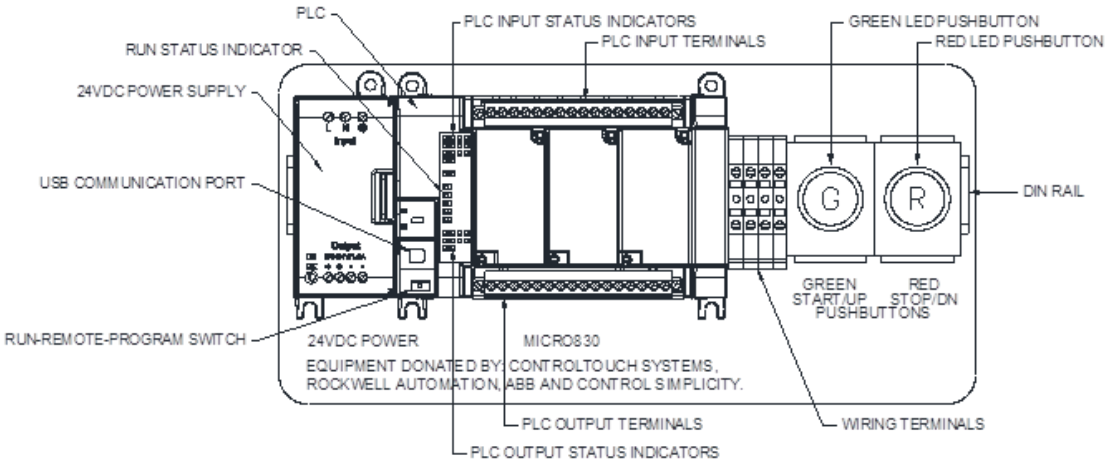

Figure 2: Allen Bradley PLC Diagram

By using both platforms, the Cornerstone project has the students use two different programming environments and have them communicate via circuitry. By having the students use the two different programming environments, the students have the opportunity to see different variations of programming software that may be encountered in industry.

Student Programming Surveys

The 2019 Spring semester students in the ENGR 111 course were surveyed at the end of the course. In the end of the semester survey there were four questions related to programming. These quantitative survey questions are shown below as well as what type of response was expected from the students:

- Which of the two programming methods used in ENGR 111 are you most comfortable interpreting?
    - Allowed answers were Arduino or PLC.
- Express your preference pertaining to the two programming methods in the course.
    - Allowed answers were one of the following five (PLC Only, Arduino Only, Arduino Over PLC, PLC Over Arduino, Both).
- Rate your current confidence level in basic programming (using any language/software).
    - 5-point Likert scale (Not Confident at All, Slightly Confident, Somewhat Confident, Very Confident, Extremely Confident).
- Rate how helpful your ENGR 110 programming instruction was at preparing you for ENGR 111 programing experience.
    - 5-point Likert scale (Not Helpful at All, Slightly Helpful, Somewhat Helpful, Very Helpful, Extremely Helpful).

These questions were self-answered by the students, and 443 students participated in the survey. The student responses are shown in Figures 3-6.

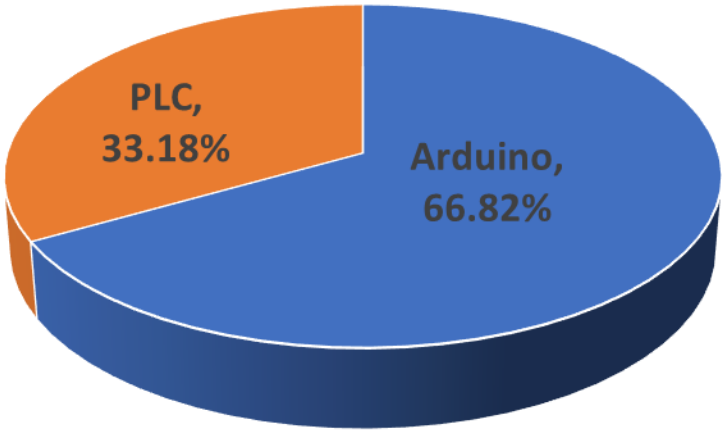

Figure 3: Which of the two programming methods used in ENGR 111 are you most comfortable interpreting?

As Figure 3 shows, two-thirds of the students said they were more comfortable with Arduino programming vs PLC programming. The authors believe this is due to Arduino syntax being similar to Python which the students were first exposed to in ENGR 110.

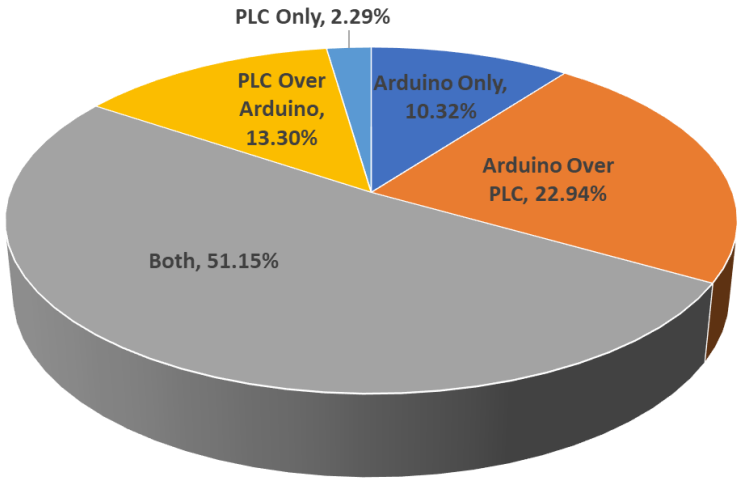

Figure 4: Express your preference pertaining to the two programming methods in this course.

However, the second survey question, Figure 4, shows that approximately half the class did not prefer one of the programming methods to the other based on the "Both" answer. Based on the first survey question it also makes sense that "Arduino Over PLC" was the second highest response to this question.

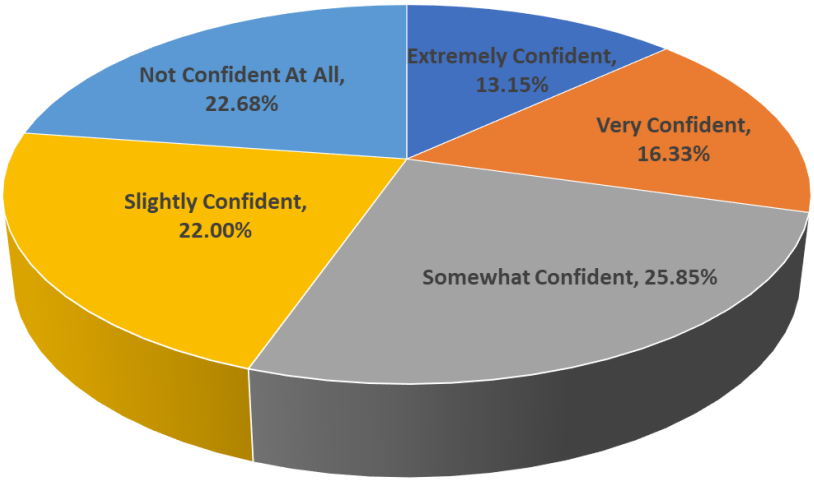

Figure 5: Rate your current confidence level in basic programming (using any language/software).

Figure 5 shows that more than half of the class (~55%) feel "Somewhat Confident" or better in their confidence level in basic programming. The authors feel that this is due to being exposed to three different programming languages over two semesters. This should help the students to realize that programming is based on structure and logic more than language syntax.

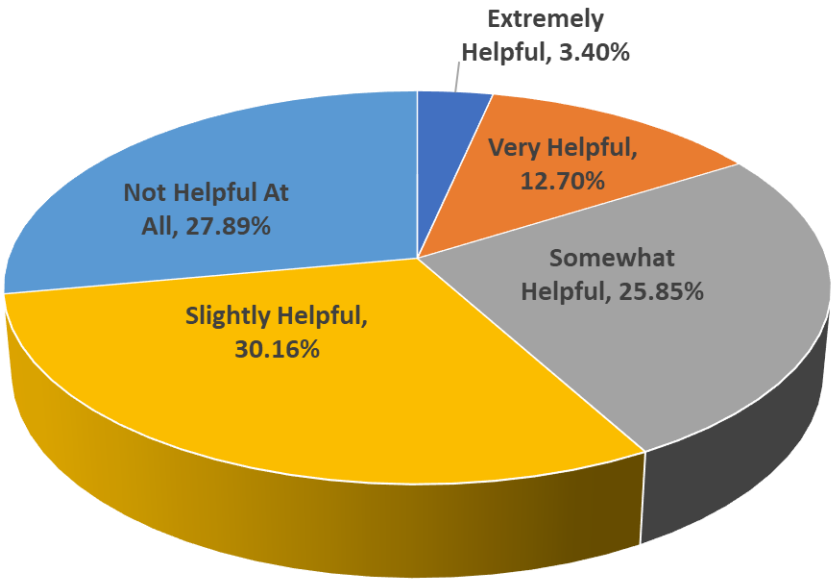

Figure 6: Rate how helpful your ENGR 110 programming instruction was at preparing you for your ENGR 111 programming experience.

The last programming survey question (Figure 6) seems to be in direct conflict with the previous answer shown in Figure 5. With more than 50% being negative ("Not Helpful At All" or "Slightly Helpful"), this suggests that many students are failing to make the connection between ENGR 110 programming pedagogy versus the programming application they are exposed to in ENGR 111.

Conclusions and Future Work

Overall the instructors and authors have been pleased with the programming results from the ENGR 111 course. It is not surprising that most students exhibited a preference in Arduino over PLC since algorithmic programming (Python and Arduino) is the methodology that they had more exposure. Although the objective is to maximize the number of students that state they are at least somewhat confident in their level of basic programming, the authors are encouraged that over half of the students expressed this level of confidence in the survey. The course sequence is not designed to turn each student into expert programmers, but the sequence is designed to expose them to programming and help them understand the basic concepts and logic involved in programming so that they will be better prepared for future courses and/or employment that require programming knowledge.

As with most first-year courses there is always room for improvement. The instructors need to do a better job helping students understand the connections between varying programming languages. The survey question in Figure 5 shows the students are comfortable with their basic level of programming knowledge, yet Figure 6 shows they are not equating the basic understanding with being exposed to multiple languages over the course of two semesters.

References

[1] V. G. Agelidis, "The future of power electronics/power engineering education: Challenges and opportunities," *IEEE Power Electron. Educ. Work. 2005*, vol. 2005, pp. 1–8, 2005, doi: 10.1109/PEEW.2005.1567584.

[2] L. Guo, "Design Projects in a Programmable Logic Controller (PLC) Course in Electrical Engineering Technology," *Technol. Interface J.*, vol. 10, no. 1, pp. 1523–9926, 2009.

[3] R. V. Krivickas and J. Krivickas, "Laboratory instruction in engineering education," *Glob. J. Eng. Educ.*, vol. 11, no. 2, pp. 191–196, 2007.

[4] "Arduino." [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3.

[5] Allen-Bradley, "User Manual - Micro830 Programmable Logic Controller Systems." [Online]. Available: https://literature.rockwellautomation.com/idc/groups/literature/documents/um/2080-um002_-en-e.pdf.

[6] R. Automation, "Connected Components Workbench Software Overview." [Online]. Available: https://www.rockwellautomation.com/en_NA/detail.page?docid=4740cbbe63a3b91551682bc7b0c6352d.