Paper ID #36361

Full Paper: First-Year Computing Course with Multiple Computing Environments - Integrating Excel, Python and MATLAB

Dr. Sean P Brophy, Purdue University at West Lafayette (COE)

Dr. Sean Brophy is a learning scientist, computer scientist and mechanical engineer with expertise in developing and research effective learning environments. His research centers on developing engineering students' expertise to adapt to new problem solving contexts.

Dr. John H Cole, Purdue University

John H. Cole (S'10–M'12) received the B.S.E.E. and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2005 and 2011, respectively. In 2013, he served as an Adjunct Professor with the American University of Kuwait and the Gulf University of Science and Technology. He is currently a lecturer at with Purdue University. His research has been concerned with power and energy systems, electromechanical energy conversion devices, modeling and simulation and engineering education.

Mr. Srinivas Mohan Dustker, Purdue University at West Lafayette (COE)

Srinivas Dustker is a Ph.D. student in Engineering Education at Purdue University. His research interests include community engaged learning, integration of service-learning in engineering curriculum, faculty development, curriculum development, education policy and technology integration in engineering education. He received his Bachelor of Engineering in Industrial Engineering and Management from B.M.S. College of Engineering, Bengaluru, India and his Master of Science in Industrial and Operations Engineering from University of Michigan, Ann Arbor, United States.

Dr. William "Bill" C. Oakes, Purdue University at West Lafayette (COE)

William (Bill) Oakes is a 150th Anniversary Professor, the Director of the EPICS Program and one of the founding faculty members of the School of Engineering Education at Purdue University. He has held courtesy appointments in Mechanical, Environmental and Ecological Engineering as well as Curriculum and Instruction in the College of Education. He is a registered professional engineer and on the NSPE board for Professional Engineers in Higher Education. He has been active in ASEE serving in the FPD, CIP and ERM. He is the past chair of the IN/IL section. He is a fellow of the Teaching Academy and listed in the Book of Great Teachers at Purdue University. He was the first engineering faculty member to receive the national Campus Compact Thomas Ehrlich Faculty Award for Service-Learning. He was a co-recipient of the National Academy of Engineering's Bernard Gordon Prize for Innovation in Engineering and Technology Education and the recipient of the National Society of Professional Engineers' Educational Excellence Award and the ASEE Chester Carlson Award. He is a fellow of the American Society for Engineering Education and the National Society of Professional Engineers.

Dr. Carla B. Zoltowski, Purdue University at West Lafayette (COE)

Carla B. Zoltowski is an assistant professor of engineering practice in the Elmore Family School of Electrical and Computer Engineering and (by courtesy) School of Engineering Education, and Director of the Vertically Integrated Projects (VIP) Program within the College of Engineering at Purdue. Prior to her appointment in ECE, Dr. Zoltowski was Co-Director of the EPICS Program. She holds a B.S.E.E., M.S.E.E., and Ph.D. in Engineering Education, all from Purdue. Her research interests include the professional formation of engineers, diversity, inclusion, and equity in engineering, human-centered design, engineering ethics, and leadership.

Full Paper: First-Year Computing Course with Multiple Computing Environments - Integrating Excel, Python and MATLAB

Introduction

Computers are a fundamental part of the engineering landscape. All engineering disciplines use computers and computing tools to model potential design solutions, collect and analyze data, create new parts through computer aided design and 3D printing and control a wide range of machinery. Computing and computer tools are a modern foundation for engineering education. Earlier studies have shown that most first-year engineering programs include programming or computer tools courses in their first-year curriculum [1]. Many challenges occur in teaching computing and computer tools in first-year engineering education courses. Students' preparation and prior experience vary significantly. Students demonstrate difficulty learning the concepts in computing and applying those concepts to writing code in a specific language [2][3]. For engineering students, there can be a disconnect between the learning outcomes desired by instructors and students' perception of the connection of writing code to their future profession. This disconnect can impact engineering students' performance to write code. One of our major learning outcomes for our students is to see the benefits of general computational skills needed in engineering problem solving. Also, they need to become proficient with computational tools and their development environments like Excel, Python and MATLAB.

Choosing the right programming language can be a challenge. Tools such as spreadsheets, including Excel, are useful for engineering analysis and design. MATLAB has been a popular choice in first year engineering programs including text based programming and graphical capabilities [4-6]. At Purdue University, MATLAB has been used in many different disciplines and the first-year experience is a key opportunity to prepare them for their majors. Python has been increasing in popularity and applications because of its low learning curve, range of application, flexibility, large open source community and it is free [7][8]. This paper describes how a team of faculty designed and implemented a first-year computing learning experience that expose students to multiple languages and tools while teaching the fundamentals of computing. The intent is to provide a learning trajectory for computational thinking [9] that help students learn how to adapt to new computer tools that automate the computational tasks. The semester starts with a short Excel experience and moves to Python and then to MATLAB. The approach uses a spiraled scaffolding instructional approach through the semester. That is, the sequence of learning experiences iteratively refines students' conception of how to design a computational system to support their engineering decision making.

Refining a First-Year Program's Computational Experience

At Purdue University all engineering students enter the college into a common first-year program. Students take Math (Calculus I and II), Science (Chemistry I, Physics I), options of Chemistry II, Biology or Computer Science (C programming), English composition, Communication and two introductory courses related to engineering design and analytical problem solving with computational tools. After completing the first-year requirements, students can transition into the major of their choice with a minimum grade point average defined by each major.

Three pathways or course sequences are used to meet the learning outcomes as shown in Figure 1. Most students take the traditional path consisting of two 2-credit introductory engineering courses. The first course, ENGR 131, focuses on design, teaming and career exploration using Excel as a computer tool to model systems and analyze data. The second course, ENGR 132, introduces students to MATLAB and more formal constructs associated with computation.

Two other pathways are offered to students to meet the first-year engineering requirements. One pathway combines the traditional introductory courses into one 4-credit course, ENGR 130. This pathway is ideal for students who have some or most of the first-year requirements met through AP credit.

The other pathway leverages Learning Communities(LC), where students participate on vertically integrated project teams. The Engineering Projects in Community Service (EPICS) program engages students in community-based design teams for one or two credits per semester. The other program, Vertically Integrated Projects (VIP), engages students in research experiences, industry sponsored projects and competition-based projects, also can be taken for one or two credits. In this pathway, the design and teamwork aspects of the introductory engineering courses are developed within the EPICS or VIP courses as students participate over two semesters but do not formally address the computation foundation. The computation requirements are assessed as part of ENGR 133 course.

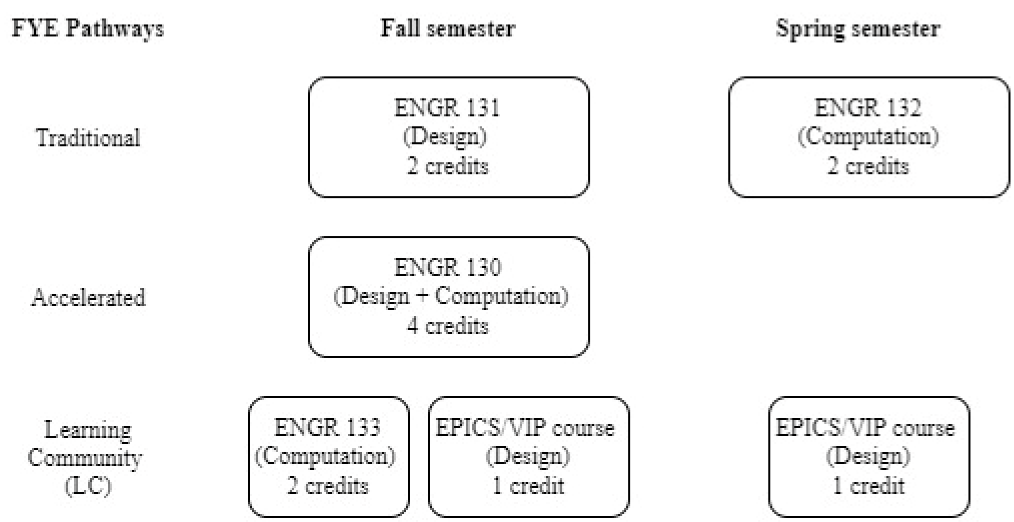


Figure 1. Pathways to meet first-year engineering credits at Purdue University

Theoretical Framework and Course Structure

The two alternate pathways share a common computational instructional sequence to develop skills using Excel, Python and MATLAB. The central premise of the instructional design is to iteratively refine students' understanding of fundamental programming constructs. The primary constructs involve data types, sequential coding, flow control with conditional expression and repetition structures, user defined function and input/output management. The theoretical framework used when developing this course was Bruner's spiral curriculum learning theory, also known as iterative learning theory [10]. The premise of a spiral curriculum involves the systematic revisiting of major concepts throughout the course/semester. At each successive iteration, the depth of the conceptual understanding increases, thus, building on the previous experience, eventually accomplishing all the learning objectives of the course. Both ENGR 130 and ENGR 133 courses use the same computational content and schedule. It begins, as shown in Table 1, with Excel as a computational tool and most students have some previous experience with spreadsheets. The start with Excel allows the students to learn the learning management system, Brightspace, and the rhythm of the course. Concepts are introduced including calculation schemes, presenting data, and conditional structures.

Table 1. ENGR 133 course structure

No.	Module	Topic	No.	Module	Topic
1	Excel 1&2	Introduction, conditionals	8	MATLAB 1	Introduction
2	Excel 3	Input/output, regression	9	MATLAB 2	Conditionals and loops
3	Python 1	Introduction	10	MATLAB 3	Functions
4	Python 2	Conditionals and loops	11	MATLAB 4	Data and file handling
5	Python 3	Functions	12	MATLAB 5	Regression
6	Python 4	Data and file handling	13	Individual Project	
7	Python Team Project				

It shifts to Python which for many of the students is their first programming language. Python can have a shallow learning curve making it accessible to a range of programming experience levels. The course uses the ANACONDA to manage the installation of the python development system. SPYDER (Scientific PYthon Devlopment EnviRonment) integrated development environment (IDE) has an easy to use interface for code development and debugging. The added benefit to the Spyder IDE is how it mirrors the MATLAB IDE used later in the semester.

Students work through a progression of learning modules which introduce basic computational constructs implemented in Python. They work in teams on assignments in class that are broken into team and individual assignments. Team assignments are designed to practice the skills needed for the individual assignments. Teams are formed to have a diversity of prior computational experience and programming expertise. The intent is that students who are still learning to program have the support of their more experienced peers. And the more experienced peers deepen their understanding as they help others learn. A team project completes the Python series and is designed to be able to be broken into components. Each team member can be responsible for a component which could be aligned with different levels of expertise.

The sequence moves to MATLAB which has been the traditional introductory programming environment for more than 25 years. The user interface is similar to the Spyder IDE and the new syntax is introduced. The same fundamental computing topics are repeated from Python to reinforce the computational and logical concepts. A goal of the course is for students to learn how to learn a new tool or environment using help functions and various resources. The move to MATLAB also provides a pause in the pace of the course with the first MATLAB week's goal focused on becoming accustomed to the MATLAB environment. It provides a lighter week before the heavier MATLAB assignments.

The culmination is a final project where each student proposes an idea they are interested in creating code to solve. They submit the proposal to the instructional team who reviews it for appropriate scope and uniqueness. Complex topics are encouraged to be broken into components and simpler ideas have additional tasks added. Students also get to choose either Python or MATLAB programming language for the project.

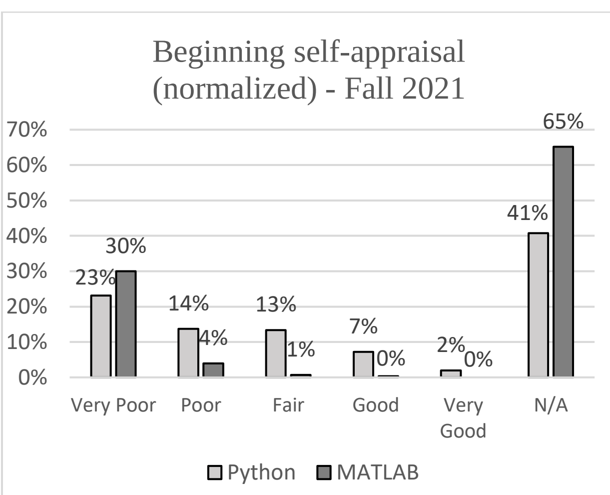
Evaluations

A goal for these courses is to increase students' abilities and confidence in computational concepts and programming to solve complex engineering tasks. A pre-semester background survey established a baseline of students' perceptions of their computational abilities. Each course had similar survey questions containing Likert scales to rate their agreement with various statements about their programming experiences [11]. These background surveys were also used to help form teams. Various assessments have been done with the class to measure students understanding of concepts and skills. Consistent feedback from students has been that this first semester class is one of the hardest or the hardest of their first semester. Previous students who return as peer teachers report that it was the hardest until they got to their sophomore engineering classes. Despite this, the majority of the students earn a grade of A and the low DFW rate is shown in Table 2. The grades are defined by individual performance on absolute scale and doesn't force into a normal distribution around letter grades. This encourages students to work and grow together so everyone can potentially earn the maximum allowable grade.

Table 2. DFW rates for ENGR 133 and ENGR 130 course in Fall 2020 and Fall 2021

Year	Course	DFW rate	Total Students	Year	Course	DFW rate	Total Students
Fall 2020	ENGR 133	2.6%	381	Fall 2021	ENGR 133	8.1%	444
	ENGR 130	2.1%	48		ENGR 130	3.9%	77

A pre-survey with response rate of 303 shows that most students do not have confidence in the programming environments that are used in the First-Year Engineering Program. About 40% have not used MATLAB and over 60% have not used Python.



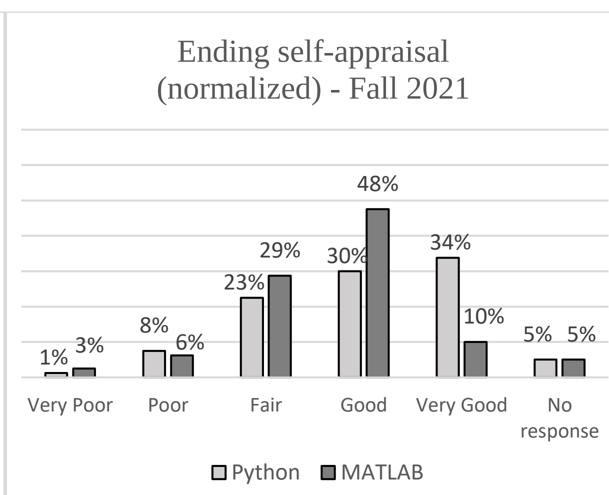


Figure 2. Beginning and ending self-appraisal survey (normalized) for Fall 2021 semester

At the end of the semester, a similar question was asked with response rate of 255. Over 60% of the students appraised themselves as good or better in Python and close to 60% of the students appraised themselves as good or better in MATLAB. The detailed breakdown of responses to the beginning and ending self-appraisal question is represented in Figure 2. Figure 2 illustrates a significant shift in the self-appraisal in programming with both Python and MATLAB languages.

In the end of semester survey, we also asked the students to rate their confidence with respect to being able to use the programming skills developed in this course in the future, in both their major and professional career. The responses to this question were linked to their gender and Figure 3 shows the end of semester self-confidence for female and male.

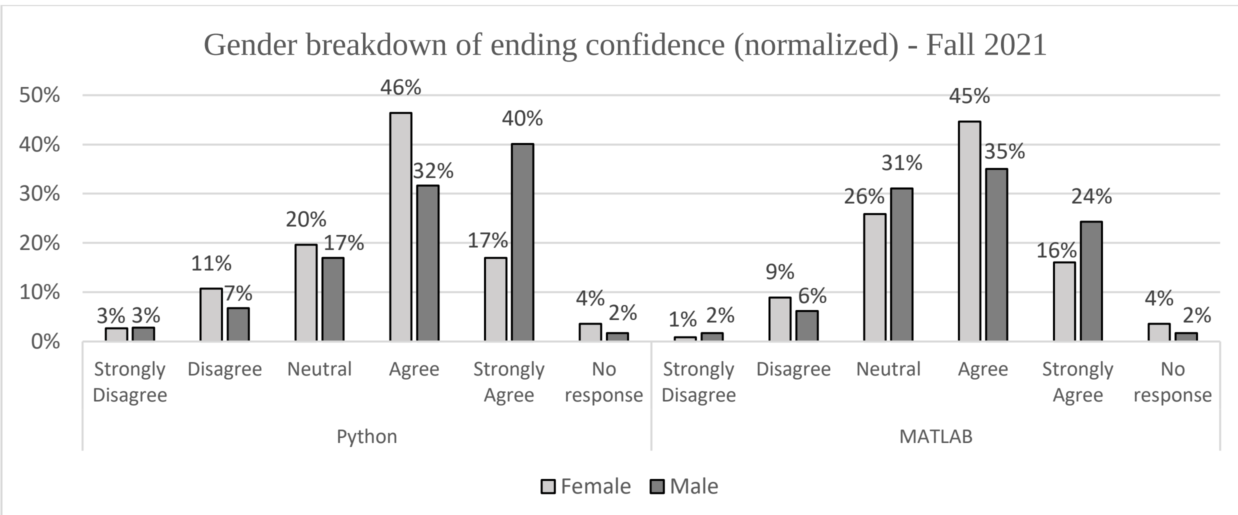


Figure 3. Ending confidence survey (normalized) for Fall 2021 semester

Discussion and Conclusion

The computing sequence of Excel to Python and then to MATLAB has been successfully used across two different first-year engineering courses to expose students to three different computer tools. Low failure rates, and significant increases in students' confidence to generate code indicate the course sequence has potential to support the claim that the approach is working. The intense pace does create some stress for the students that has to be managed. Data shows that most students who stick with the courses and do the assignments do very well. Problems arise when they disengage with the learning experiences. Instruction includes encouraging them and discussing the challenges of learning computing. The introduction of computational topics in Python provides a foundation for conceptual understanding of fundamental programming constructs. Students demonstrated they can refine this conceptual understanding when they revisit the concepts using MATLAB. Students adapted to the rapid pace of learning activities during the MATLAB sequence. However, an important learning principle is to schedule a light week after the team project in Python. This provides the benefits for students to have a mid-semester break and a gentle introduction to the second programming language.

The individual projects offer a final assessment. Students are very motivated for the final project, and many describe it as their favorite part of the course. They often also report that some aspect of the computing concepts made sense after completing their project. Students' perceptions of programming experience level increased significantly. Female students were not as confident at the end of the semester, which may be due to lower programming experience before the course compared to their male counterparts. Future work will be done to better understand this potential gender difference. Additional efforts are underway to produce learning materials to support students' exploration and practice of key computational constructs including instructional videos and programming exercises.

References

[1] K. Brannan and P. Wankat, "Survey of first year programs", in *Proc. 2005 ASEE Annual Conference & Exposition*, Jun. 2005, pp. 10.1188.1 – 10.1188.23.

[2] M. McCracken, et al. "A multi-national, multi-institutional study of assessment of programming skills of first-year CS students", *Working group reports from ITiCSE on Innovation and technology in computer science education*, 2001, pp. 125-180.

[3] L. Thomas, M. Ratcliffe, J. Woodbury and E. Jarman, "Learning styles and performance in the introductory programming sequence", *ACM SIGCSE Bulletin 34,* 1, 2002, pp. 33-37.

[4] P.T. Goeser, T. Murphy, C. Williams, D. Calamas and J. Choi, J, "MATLAB Marina: The Primary Resource for MATLAB in a Freshmen Computing Applications for Mechanical Engineering Course", in *2021 ASEE Virtual Annual Conference Content Access*, 2021.

[5] A. Akundi, T. L. B. Tseng, Z. Cao, and H. Kim, "A Deep Learning Graphical User Interface Application on MATLAB", in *2018 ASEE Annual Conference & Exposition*

[6] D. Belfadel, M. Zabinski and I. Macwan, "Introduction to MATLAB Programming in Fundamentals of Engineering Course", in *2021 ASEE Virtual Annual Conference Content Access*.

[7] B. J. Furman, S. Ahsan and E. Wertz, "Making the Move from C to Python With Mechanical Engineering Students", in *2020 ASEE Virtual Annual Conference Content Access*.

[8] E. Ebrahimzadeh and N. M. Safai, "Should "Python for Engineers" be a Course Taught to Freshmen Engineering Majors in the USA and Abroad?", in *2019 ASEE Annual Conference & Exposition*.

[9] S. B. Brophy and T. A. Lowe, "A learning trajectory for developing computational thinking and programming", in *School of Engineering Education Faculty Publications*, 2017.

[10] J . Bruner, "*The Process of Education*". Cambridge, MA: The President and Fellows of Harvard College, 2009.

[11] R. Likert, "A technique for the measurement of attitudes", *Archives of psychology*, 1932.