Teaching Programming in the Context of Solving Engineering Problems

Joseph P. Hoffbeck, Heather E. Dillon, Robert J. Albright, Wayne Lu, Timothy A. Doughty
Shiley School of Engineering
University of Portland
Portland, Oregon, USA

Abstract—**This paper describes the course that was developed at the authors' University to introduce all first-year engineering students to the fundamentals of computer programming within the context of solving engineering problems. This two credit-hour, semester-long course incorporates the programming language MATLAB and is a required course for students who major in civil, electrical, and mechanical engineering. The course, which is usually taken during the second semester of their first year of study, was designed to utilize active learning techniques by having the students complete a series of laboratory exercises and projects that introduce computer programming and engineering applications.**

This paper describes the origins of the course, the laboratory exercises and projects, how the course was administered, and an assessment of how successful the course was based on student grades, student feedback, and a student survey. The results indicate that the course increased students' knowledge of programming in the context of solving engineering problems.

Keywords—computing, engineering, MATLAB, education

I. Background

MATLAB® is a user friendly high-level computing tool widely used in academia and industry for data analysis, algorithm development, simulation and design prototyping. Due to MATLAB's programming language syntax, it can be taught as a programming language and, oftentimes, the only programming language for non-ECE (Electrical and Computer Engineering) engineering disciplines. MATLAB also includes various useful toolboxes specialized in different applications. Because MATLAB's wide spread use in engineering curricula, many universities offer a course to teach MATLAB in their freshman-year engineering programs so that first-year students can learn basic programming concepts and practical engineering problem solving skills.

Some universities use a module-based approach to teach MATLAB programming skills. The coverage of these MATLAB modules varies from a few hours to one-third or one-half of a semester [1-5]. Some universities teach MATLAB in an introductory programming language course with C++ or LABVIEW [6,7]. Some universities integrate MATLAB programming into an existing course or just teach MATLAB in specific engineering disciplines [8-11].

There are many universities that teach MATLAB in a dedicated MATLAB course ranging from 1 credit-hour to 3 credit-hours to first-year engineering students. Some courses integrate hardware-based data acquisition equipment to collect data and use MATLAB to process the data [12,13]. Most of the MATLAB courses incorporate group-based projects so that students can apply MATLAB to solve open-ended engineering problems [14-19].

To ease the pressure on undecided freshman students to decide a major by the end of the first semester and to enable smooth transfers between different majors, a common first-year curriculum for all engineering students (civil engineering, electrical engineering and mechanical engineering) has been established in the School of Engineering at the authors' University starting with the 2013-2014 academic year. To consolidate the three courses teaching MATLAB in different disciplines in the School of Engineering, a new 2 credit-hour MATLAB-centered course, EGR 111 Engineering Computing with Applications, was offered for the first time in the Spring Semester 2014. The main features of the EGR 111 course are 1) multidisciplinary students from three engineering disciplines; 2) emphasis on group learning of the MATLAB programming language for the first half of the semester; and 3) implementation of three open-ended group projects related to each discipline in the second-half of the semester.

The most innovative feature of the EGR 111 course is the implementation of three open-ended group projects related to each discipline in the School of Engineering. This feature, which is not found in the listed references, allows the students to develop their programming skills while solving engineering problems in three different engineering disciplines. The EGR 111 course content is discussed in the following sections.

II. Description of the Course

The authors' University is a small private comprehensive university that offers degrees in civil, electrical, and mechanical engineering. During their first semester, the engineering students take a required introductory engineering course where they work in small groups on a design project to learn basic design methodology, to practice teamwork, technical writing, public speaking, and learn about the different disciplines in engineering.

In the second semester of their first year, the students take a required introductory applied programming course, which was offered for the first time in Spring Semester 2014. This new course introduced the basics of computer programming in the context of solving engineering problems. Part of the justification for this course, was to provide a common curriculum for all three engineering programs for the first year, which allowed students to learn more about the different disciplines in engineering before deciding on their major and made it easier for students to switch between majors in their first year. Before this course was developed, the students took a discipline-specific course during their second semester. This new common course was also designed to provide students with the basic programming and data analysis skills, but now, in addition to working in multi-disciplinary teams, students work to solve problems representing several disciplines.

The new introductory applied programming course was a two credit-hour, semester-long course that met twice per week. The class met in a computer classroom so each student had a computer with MATLAB installed. The maximum class size was 35, which was limited by the number of computers in the classroom. In spring 2016, there were six sections of the course taught by five different full-time engineering faculty members. On a typical day, the instructor gave a short introduction to the laboratory experiment, which was provided on the course website, and then the students spent the remainder of the class completing the experiment. The instructor and a qualified undergraduate student helper were present to answer students' questions and help them if necessary. In this arrangement students were actively engaged in learning the material and solving problems during the classes.

The main objectives of the course were to teach students to do the following:

- To write MATLAB programs, including assigning, indexing, plotting, importing and exporting data, looping, branching, and writing script files and user-defined functions.
- To demonstrate skills in designing, implementing, documenting, testing, debugging, and analyzing programs in the context of engineering problems.
- To apply MATLAB programming to technical problems in multiple engineering disciplines with applications such as image and sound processing, heat transfer, and stress-strain analysis.

The schedule for the course is listed in Table I. The first part of the course consisted of a series of laboratory exercises that introduced the basic features of the MATLAB programming language. These exercises can be viewed on the course website [20]. The first exercise included topics such as how to enter scalars, vectors, and matrices, how to index vectors and matrices, and use of the colon operator. The second exercise included how to load data from a file into MATLAB and how to plot the data. Whenever possible, real engineering data, such as the power output of a solar panel, was used in the exercises. In the third exercise, the students utilized trigonometric functions by making a plot of the height of a piston as a function of the angle of the crank. In the next exercise, the students loaded an audio file into MATLAB and played it back backward, which was amusing for the students and reinforced how to use the colon operator to index a vector. The students also generated sinusoids to synthesize a short song and used filters to remove jet noise from a recording of birds chirping. Other exercises introduced complex numbers, relational operators (such as <, >, =), conditional execution (if, else, and end commands), user-defined MATLAB functions, and loops (for and while commands).

TABLE I. COURSE SCHEDULE

Class Period	Topic
1	Course Overview
2	Introduction to MATLAB
3	Plotting Data
4	Trig Functions
5	Audio Processing
6	Complex Numbers
7	User-defined Functions and Relational Operators
8	Conditional Execution
9	Loops
10	Fourier Series
11-12	Review for Midterm Exam and Exam
13	Image Processing
14	Heat transfer
15	Stress-strain
16-18	Project 1 (Traffic flow)
19-22	Project 2 (Heart rate)
23-26	Project 3 (Projectile motion)
27-28	Review for Final Exam and Exam

The next four exercises were engineering applications: Fourier series, image processing, heat transfer, and stress-strain. In the Fourier series lab, the students used loops to add up a Fourier series to see how it converged to a square wave or triangle wave, and added up the harmonics to synthesize the sound of a trumpet.

In the image processing exercise, the students used MATLAB to perform basic manipulations of black and white images such as changing pixel values, brightening, rotating, or superimposing two images. The students were also given an image that was deliberately modified to obscure the scene (see Fig. 1) and were asked to write a program to undo the modifications to reveal the original image (see Fig. 2). This exercise required nested for loops, if statements, and indexing. There were also optional activities to manipulate color images.

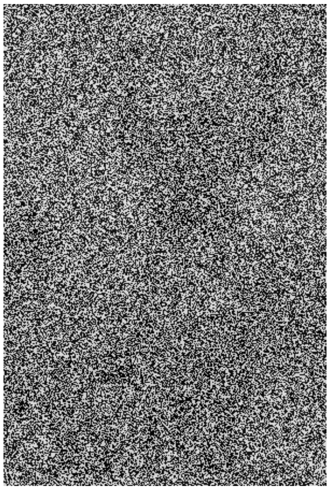

Fig. 1. Obscured image that students used to test image processing.

Fig. 2. Output image that students discover when the image processing is complete.

In the heat transfer exercise, the students wrote a program to compute the temperature of a 1-dimensional bar over time when the temperatures of the ends of the bar are held constant. This lab used nested for loops, indexing, and plotting.

In the stress-strain experiment, the students were given real force and strain data from a load frame, and they computed and plotted the stress-strain curve for the material.

Most of the second half of the course consisted of three projects, each of which focused on a different discipline in engineering. The goals of the projects were to have the students apply the programming skills that they learned to solve interesting problems that were related to civil, electrical, and mechanical engineering. Three or four class periods were devoted to each project, whereas each laboratory exercise was completed in a single class period.

In the first project, which focused on an application in civil engineering, pairs of students observed the traffic on a local street adjacent to the campus and counted the number of vehicles that passed in 15 minute intervals. The data from all students was collected over an 8 hour period and made available to them. The students were then asked to write a MATLAB program to load in the data and to compute the peak hour factor for the street. The program needed to replace missing data with the value from the previous 15 minute interval, plot the number of vehicles for all of the 15 minute intervals, and compute the peak hour factor for the street. This project required if statements, loops, and plots.

The second project, which related to an application in electrical engineering, was adapted from the IEEE Real World Engineering Project [21]. The goal of this project was to write a MATLAB program to analyze real electrocardiogram (ECG) signals to determine if they were dangerous, life-threatening ventricular tachycardias (VTs) or more benign supraventricular tachycardias (SVTs). See Fig. 3 and 4 for an example of each signal. The students' programs needed to calculate the onset time, which involved finding the first peak in the derivative of the average of three peaks in the ECG signal. The onset time was calculated for some known example signals, which allowed the students to find a threshold that could separate the two types of signals. Then the onset time of the unknown signals was calculated and classified as one of the two types of signals. They compared their assessment results to those listed in the reference by a cardiologist.

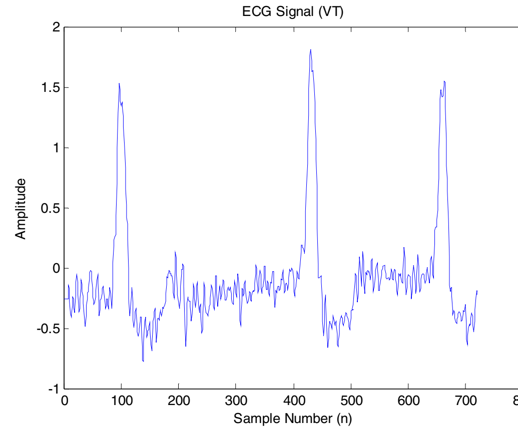

Fig. 3. ECG Signal (VT) analyzed by students.

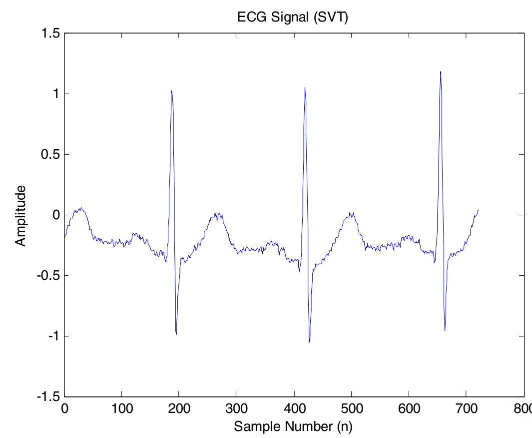

Fig. 4. ECG Signal (SVT) analyzed by students.

The last project, which was related to mechanical engineering, had the students write a function to plot the maximum horizontal distance of a projectile given the initial velocity and angle, which illustrates the well-known result that the angle that leads to the maximum distance is 45 degrees. However, the students were then asked to write a program to plot the maximum distance when the slingshot is pulled back to the ground. In this situation, the 45 degree pull does not allow the slingshot to be pulled as far back as for a smaller angle, so the angle that leads to the maximum distance is not obvious. During one of the class periods, the students went outside and used slingshots to launch foam balls and measure the maximum distance for different angles as shown in Fig. 5. Since the foam balls have a lot of air resistance, the distances differed significantly from the theoretical values, but the data was useful in verifying the overall result. In the future, a different type of ball may be used or drag may be included in the model to reduce the difference between the theory and experiment.

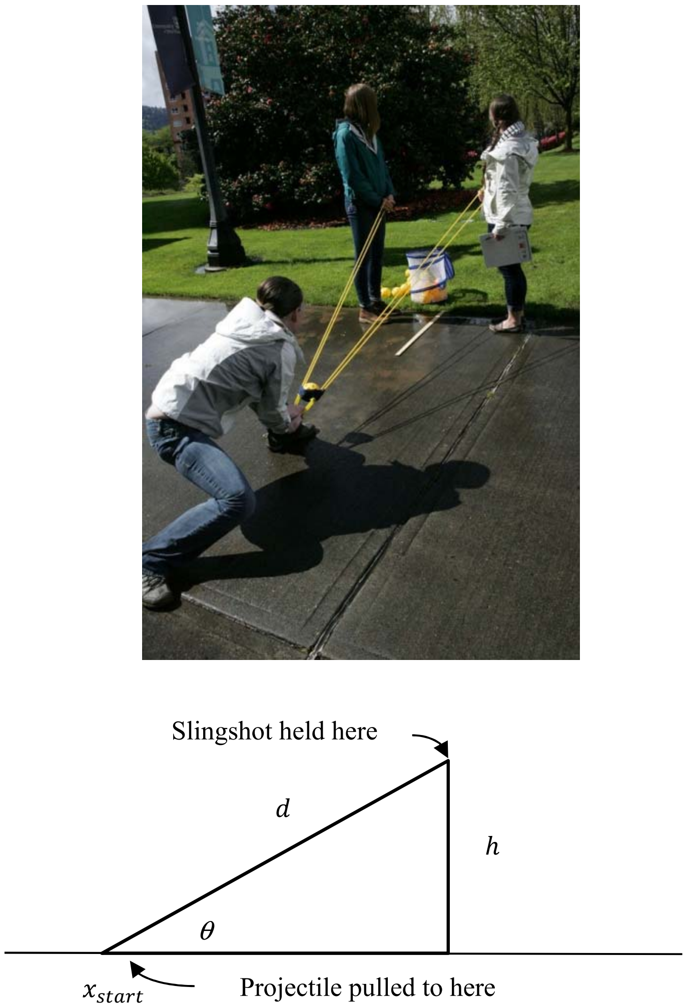

Fig. 5. Image and model of the stretched slingshot in the projectile project.

One of the assignments was to gather information about an engineering field in which they were interested and to write a short paper on it. The students gathered data from a variety of sources such as interviews, articles, and books. The paper covered topics such as educational requirements, types of work available, advantages and disadvantages of that field, the approximate salary, and other relevant information. In the paper, the students also reflected on if they felt that the field was right for them.

The course was graded on the labs (10%), the midterm exam (25%), the final exam (25%), four quizzes (15% total), homework (15%), the career assignment (5%), and participation/miscellaneous (5%).

The scores on the laboratory exercises were based on how many of the lab exercises the students successfully completed. When the students demonstrated that they had completed an exercise, their group was "checked off" and received full credit for that exercise. If they did not successfully complete the exercise, they received no credit. The homework problems required the students to write MATLAB programs similar those in the laboratory exercises, but the homework was submitted individually to Mathwork's Cody Coursework system where it was automatically graded by scripts written by the instructors of the course. The quiz and exam questions required the students to write a short MATLAB program to solve a task or to determine the output of a given MATLAB program. The quizzes and exams were on paper and did not involve running the MATLAB program during the quizzes or exams.

III. Assessment and Evaluation

In order to determine if the course was successful in achieving the course objectives, the course was assessed and evaluated using grades on the assignments and exams, anonymous course evaluations from the students, and a student survey. The final exam covered the first course objective (to write MATLAB programs, including assigning, indexing, plotting, importing and exporting data, looping, branching, and writing script files and user-defined functions). The number of questions on the final exam in the six sections were 5, 5, 9, 14, 16, and 17, and some of the questions had multiple parts. The score on the final exam, averaged across all six sections, was 83.1% (see Table II).

The second and third course objectives were to demonstrate skills in designing, implementing, documenting, testing, debugging, and analyzing programs in the context of engineering problems, and to apply MATLAB to technical problems in multiple engineering disciplines with applications such as image and sound processing, heat transfer, and stress-strain analysis. These last two objectives were covered mainly in the Laboratory Exercises and Homework/Projects, which included the three discipline-specific projects. The average score for Laboratory Exercises and Homework/Projects was 95.8% and 89.2% respectively (see Table II).

TABLE II. Average scores across the six sections

Category	Average Score (%)
Laboratory Exercises	95.8
Homework/Projects	90.9
Quizzes	81.3
Career Assignment	91.8
Participation/Miscellaneous	92.3
Midterm Exam	88.4
Final Exam	83.1
Total Score	87.9

During the last two weeks of the course before the final exam, the students filled out anonymous on-line course evaluation surveys. The evaluations had five questions where the students can respond either "Strongly Agree", "Agree", "Neutral", "Disagree", or "Strongly Disagree." For the question that asked if the students agreed with the statement "Activities and teaching methods effectively conveyed the course content," the results from the average for the six sections of the course are shown in Table III.

TABLE III. COURSE EVALUATION RESULTS FOR "ACTIVITIES AND TEACHING METHODS EFFECTIVELY CONVEYED THE COURSE CONTENT."

Response	Average Percentage of Students (%)
Strongly Agree	35.8
Agree	48.7
Neutral	8.8
Disagree	2.7
Strongly Disagree	4.2

The course evaluations also include two questions where the students can enter written comments. For the question "Describe the strengths of this instructor and course," typical responses included that the course was a good introduction to MATLAB, it was interesting but difficult, and it was good to be active rather than sitting and listening.

For the question "What changes, if any, would you suggest for this instructor and course?", typical responses included asking for more instruction before the labs, incorporating more of the material required to do the projects into the earlier laboratory exercises, allowing students to select a project related to their major, and requesting more helpers to answer questions during the class. Also, some of the students thought that the projects were too difficult.

Based on the assessment and evaluation of the first offering of the course, several changes were implemented. The primary change was to group the students into teams on the first day of the semester, which served two primary functions. The first was that students used their neighbors as resources and this has created a culture of peer instruction and interdisciplinary group work. This grouping also significantly reduced the amount of time students had to wait for assistance. In keeping with this culture, students will be sharing their career assignment reflections with their teams to facilitate a discussion about students' perceptions of themselves and their chosen majors.

Additionally, two extra lab periods were dedicated to the projects. The labs on filtering and conditional execution were absorbed into related labs, which freed an extra session for both the arrhythmia and projectile projects. This change allowed the students more time to reflect on how they can best approach the projects, rather than asking the instructors how to proceed. It is expected that this change will facilitate greater student ownership of the material in the course and of their overall education. Making this stage of reflection and planning deliberate in their first year should benefit them throughout their student and professional careers.

The final form of assessment was a student survey to evaluate how successful the students felt the course structure was in helping them learn MATLAB and confirming career choices. The survey was completed by 72 students out of 166 students taking the course in six different sections. The first question asked the students to "Consider your comfort level with MATLAB programming techniques at the beginning of the semester. How much more comfortable are you with using them now?" on a Likert scale. 82.4% of students responded with a positive Likert response, from a great deal to a moderate amount, indicating the majority of students perceive that they have learned MATLAB to some degree. The breakdown of student responses is shown in Fig. 6.

To further understand what parts of the introductory programming class helped the students most, the survey asked the students to rank different class activities. The Likert score out of 5 options was averaged and the results are shown in Fig. 7. The students reported that homework, office hours, and exam/quiz study was the most helpful for them in learning MATLAB skills, however other facets of the class were ranked nearly as helpful.

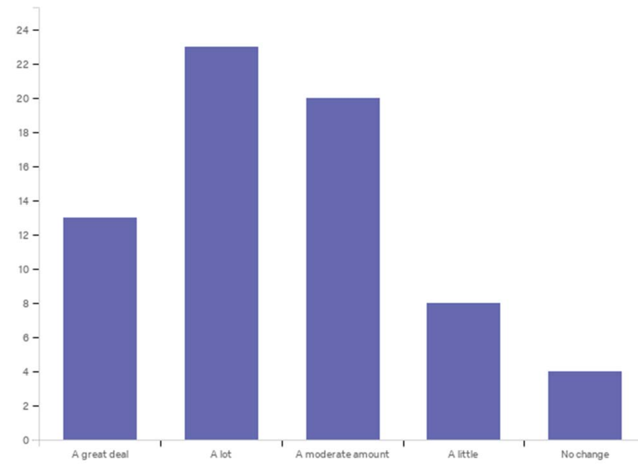

Fig. 6. Survey results for the question "Consider your comfort level with MATLAB programming techniques at the beginning of the semester. How much more comfortable are you with using it now?"

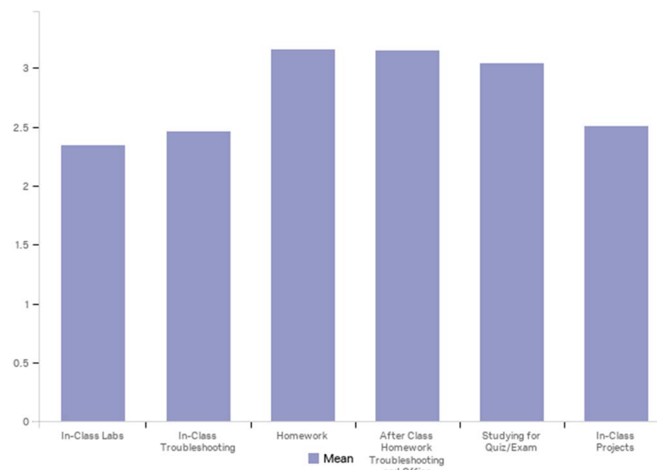

Fig. 7. Survey results for the question "We have used a range of methods to help you learn the material in this class. What methods have been the most useful to you in learning the programming techniques in this class?" Each bar represents the average student response for each topic area.

To understand what impact the class structure might have on the student career choices the survey asked "Consider your comfort level with your choice of discipline (major) at the beginning of the semester. How much more comfortable are you with your choice now, even if it has changed?" As shown in Fig. 8 the students reported this as less impactful, with 70.6% of students indicating the comfort level with a specific major had improved from a great deal to a moderate amount. This result may indicate that the students already had a high comfort level with choice of major before starting the class. This is reasonable since the introduction to engineering course taken by students in the prior semester focuses extensively on career options. However, for the students who reported that the class did make them more comfortable with choice of major, this is a very helpful outcome as they enter the second and more difficult year of engineering coursework.

To further understand what parts of the introductory programming class helped the students explore career options, a question was added to let students rank specific activities. The results are shown in Fig. 9. The students reported that the career essay assignment was the most helpful, but other labs and projects with discipline focus were also beneficial with mean Likert scores over 3.

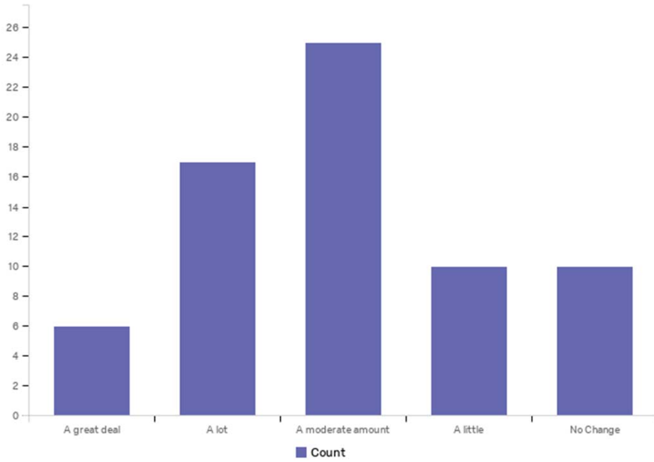

Fig. 8. Survey results for the question "Consider your comfort level with your choice of discipline (major) at the beginning of the semester. How much more comfortable are you with your choice now, even if it has changed?"

IV. Conclusions

An introductory applied programming course was described that used the programming language MATLAB in the context of addressing engineering problems. The course consisted mainly of a series of laboratory experiments which led up to three discipline-specific projects which students completed in multi-disciplinary teams. The primary objectives were assessed and evaluated using the students' grades, anonymous course evaluations, and a survey. The assessment results indicated the course was effective at helping students learn introductory programming techniques.

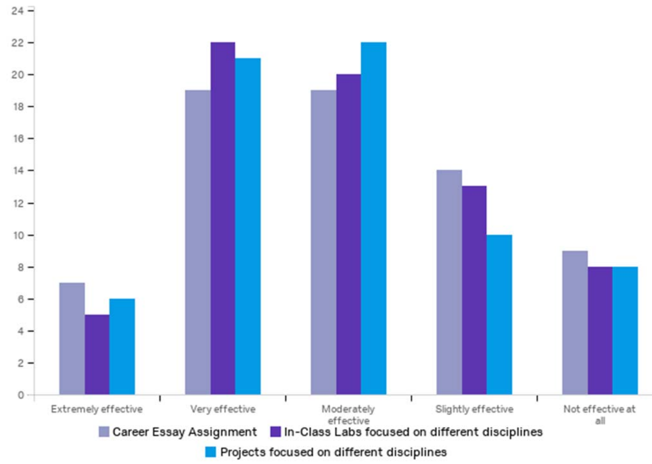

Fig. 9. Survey results for the question "We have used a range of methods to help you learn more about possible engineering career disciplines. What methods have been the most useful to you in learning about engineering disciplines?" The count for student responses in each topic are shown.

Acknowledgment

The authors wish to thank the students who participated in this course during the study.

References

[1] Laplume, P. and Ruane, M., "Using Imaging to Introduce Engineering to Freshmen", *Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition*

[2] Firebaugh, S., Salem,T., Etter, D., Shenck, N. and Voigt, R., A "Hands-On" Module to Introduce Freshmen to Electrical Engineering, *Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition*

[3] Bualuan, R., "Teaching Computer Programming Skills to First-Year Engineering Students Using Fun Animation in MATLAB", *Proceedings of the 2006 American Society for Engineering Education Annual Conference & Exposition*

[4] Adams, R., Hannemann, J., and Holloway, L., "Creativity and Design: A General Education Course for ECE Freshman", *Proceedings of the 2012 American Society for Engineering Education Annual Conference & Exposition*

[5] Howe, C., and Blandford, D., "Frequency Domain Concepts for EE Freshman", *Proceedings of the 2013 American Society for Engineering Education Annual Conference & Exposition*

[6] Azemi, A. and Pauley, L., "Using Matlab to Teach the Introductory Computer-Programming Course for Engineers", *Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition*

[7] Hrynuk, J., Pennington, M., Illig, D., and Dempsey, J., "Freshman Engineering: An Introductory Computer Course Teaching MATLAB and LabVIEW", *Proceedings of the 2008 American Society for Engineering Education Annual Conference & Exposition*

[8] Devens, P., "MATLAB & Freshman Engineering", *Proceedings of the 1999 American Society for Engineering Education Annual Conference & Exposition*

[9] Forsberg, C., "Experiments for a Freshman Engineering Computing Course", *Proceedings of the 2013 American Society for Engineering Education Annual Conference & Exposition*

[10] Morrell, D., "Design of an Introductory MATLAB Course for Freshman Engineering Students", *Proceedings of the 2007 American Society for Engineering Education Annual Conference & Exposition*

[11] Bean, J., Dempsey, J., French, S., Hrynuk, J., Illig, D., O'Loughlin, V., Pennington, M., Zander, A. and Zenger, G., "Defy Convention: Past

Students Overhaul Hands On Freshman Computer Engineering Course", *Proceedings of the 2007 American Society for Engineering Education Annual Conference & Exposition*

[12] Pritchard, J., and Mina, M., "Hands-on, discovery, critical thinking, and freshman engineering: A systems level approach to learning and discovery", *Proceedings of the 2012 American Society for Engineering Education Annual Conference & Exposition*

[13] Hamrick, T. and Hensel, R., "Putting the Fun in Programming Fundamentals - Robots Make Programs Tangible", *Proceedings of the 2013American Society for Engineering Education Annual Conference & Exposition*

[14] Hensel, R. and Sun, Y., "Application of Object Scaffolding to Develop A Hands-On, Problem-Centered, and Project-Based Freshman MATLAB® Course", *Proceedings of the 2006 American Society for Engineering Education Annual Conference & Exposition*

[15] Morrell, D., "Design of an Introductory MATLAB Course for Freshman Engineering Students", *Proceedings of the 2007 American Society for Engineering Education Annual Conference & Exposition*

[16] Goeser, P., Johnson, W., Bernadin, S. and Gajdosik-Nivens, D., "Work-in-Progress: The Impact of MatLab Marina - A Virtual Learning Environment on Student Learning in a Computing for Engineers Course", *Proceedings of the 2013 American Society for Engineering Education Annual Conference & Exposition*

[17] Bucks, G., Ossman, K., Kastner, J., Boerio, J. and Torsella, J., "First Year Engineering Courses Effect on Retention and Student Engagement", *Proceedings of the 2014 American Society for Engineering Education Annual Conference & Exposition*

[18] Ossman, K. and Bucks, G., "Effect of Flipping the Classroom on Student Performance in First Year Engineering Courses", *Proceedings of the 2014 American Society for Engineering Education Annual Conference & Exposition*

[19] Morse, S., Morse, A., Uddameri, V., Hernandez, E. and Ernst, D., "The Impact of Reducing Numerical Methods and Programming Courses on Undergraduate Performance", *Proceedings of the 2014 American Society for Engineering Education Annual Conference & Exposition*

[20] EGR111 Engineering Computing with Applications Home Page, University of Portlnad, https://sites.up.edu/egr111/.

[21] IEEE Real World Engineering Projects, http://www.realworldengineering.org