AC 2011-1282: REDESIGN OF FRESHMAN ELECTRICAL ENGINEERING COURSES FOR IMPROVED MOTIVATION AND EARLY INTRODUCTION OF DESIGN

Phillip Wong, Portland State University, ECE Department

Phillip Wong received his M.S. degree in electrical engineering from Carnegie Mellon University. He is an adjunct instructor and Lab Coordinator for the Electrical and Computer Engineering Department at Portland State University.

Melinda Holtzman, Electrical and Computer Engineering Department, Portland State University

Melinda Holtzman received her PhD from the University of Nevada, Reno. She is a Senior Instructor in the ECE department at PSU.

Branimir Pejcinovic, Portland State University, ECE department

Branimir Pejcinovic received his Ph.D. degree from University of Massachusetts, Amherst. He is an Associate Professor and Associate Chair for Undergraduate Education at Portland State University, Electrical and Computer Engineering department. In this role he has led department-wide changes in curriculum with emphasis on project- and lab-based instruction and learning. His research interests are in the areas of semiconductor device characterization, design and simulation, signal integrity and THz sensors. He is a member of IEEE and ASEE.

Malgorzata Chrzanowska-Jeske, Portland State University

Malgorzata Chrzanowska-Jeske received her M.S. degree in electronics engineering from Politechnika Warszawska (the Technical University of Warsaw) in Warsaw, Poland, and a Ph.D degree in electrical engineering from Auburn University, Auburn, Alabama, USA.

Since 1989, she is with the Department of Electrical and Computer Engineering at Portland State University in Portland, Oregon, currently as Professor. She served as Chair of the Department from 2004 to 2010 and has overseen major changes in the curricula of the electrical engineering and computer engineering BS degrees. Her research interests include CAD for VLSI IC, MS-SOC, 3D ICs, nanotechnology and nano/bio systems, design for manufacturability, and design issues in emerging and renewable technologies. She has published more than 100 technical papers and has served as a panelist and reviewer for the National Science Foundation (NSF), and as a reviewer for National Research Council Canada (NRC) and many international journals and conferences.

She has served on the Board of Governors of IEEE Circuits and Systems Society since 2008. Recently, she received IEEE Council on Electronic Design Automation 2008 Donald O. Pederson Best Paper Award for her paper published in IEEE Transactions on Computer-Aided-Design of Integrated Circuits and Systems.

Redesign of Freshman Electrical Engineering Courses for Improved Motivation and Early Introduction of Design

1. Introduction and Motivation

The student experience during the freshman year has been recognized as one of the keys to not only attracting more students into engineering and improving retention, but also to forming some significant attributes of successful engineering graduates[1]. Portland State University is an urban university, and its Electrical and Computer Engineering (ECE) department serves a relatively large and very diverse student population including a large fraction of transfer and part-time students. Traditionally, all engineering disciplines within our Maseeh College of Engineering and Computer Science had a similar freshman year curriculum. The common entry course – Engineering and Applied Science (EAS) 101 – served as the cornerstone along with one or two additional courses which were more discipline specific. In ECE these two courses covered introduction to programming and digital logic, with the former taught by the Computer Science (CS) department and the latter by ECE.

There were a number of reasons why we decided to redesign our undergraduate curricula. Through our own assessment and feedback from employers and alumni, several programmatic issues were identified: a) insufficient programming skills, b) introduction to design only in upper-division courses, c) weak communication skills. At the same time, many schools across the United States were reducing the credit load in Electrical Engineering (EE) to 180 credits, and we had started feeling pressure from our students and prospective students as well. This prompted our examination into ways of rationalizing and potentially reducing the number of courses. Finally, we wanted to make our program more attractive to undecided and traditionally under-represented groups of students. We realized that solutions for many of the identified issues might be found by focusing on how we introduce freshman students to electrical and computer engineering fields.

2. Our Approach to Curriculum Development

These findings forced us to consider the redesign of our freshman courses more holistically instead of just looking for ways to modify content or delivery of one of the courses. The first realization was that we needed to take control over the freshman engineering curriculum for our students. This meant we had to offer ECE-oriented classes instead of general classes designed to cover all the traditional engineering disciplines. The second realization was that we needed a completely new course to be the gateway into the ECE program, one that would be more inviting to students. Instead of trying to filter them out of the program, we would present them with a spectrum of engineering challenges that are fun to work on[2]. To use nomenclature from Spurlin[1] our changes were on the "curriculum" level.

There is an obvious down-side to our approach, namely that students will not get any intentional "exposure" to other engineering disciplines. However, we would argue that simply providing students with tidbits of information related to other disciplines does not serve any pedagogical purpose other than providing a survey of what is available. Our new sequence of courses serve a well defined pedagogical and curriculum purpose albeit within a more narrow focus.

Given that there is a large body of research indicating that active student learning in the form of hands-on projects and lab-based approaches are very effective[3,4,5], we designed all of the courses with this in mind. Our assessment plans are largely based on direct and quantitative techniques with some student surveys providing a more indirect and qualitative feedback. In the following we will discuss design, implementation, assessment, and analysis of a three course sequence that was introduced in the 2009-10 year.

3. Course Development

Historically, the EAS 101 syllabus followed a traditional set of topics, such as problem solving and data presentation. Rudimentary coverage of mechanical systems, electric circuits, fluid mechanics, thermodynamics, and statistics was provided. Other subjects included some principles of design, engineering economics, ethics, and a very short MATLAB tutorial. The CS programming course was devoted exclusively to teaching the fundamentals of computer programming, with UNIX serving as the development environment. These courses exhibited a number of deficiencies for prospective ECE students:

- For those who were still undecided about engineering as a career path, the EAS and CS courses did little to promote the many opportunities available in the ECE field.
- While diverse engineering topics and problems were discussed, it was only at a shallow level and did not cater to the interests of ECE students. Very little time was actually spent discussing electrical engineering principles.
- Many students felt that the predominantly lecture-based approach was uninteresting and did not hold much relevance to "real-world" applications.
- There was no hands-on laboratory experience in the freshman year. Students did not experience the creative and "fun" side of engineering.
- The lack of laboratory experience and the cursory introduction to MATLAB in the first year caused some students to feel overwhelmed when those aspects were introduced rapidly in the sophomore year.

To address these concerns, we developed the following three quarter sequence as a replacement for the EAS 101 and CS classes, while keeping the third digital logic class as a separate course:

- *ECE 101 Exploring Electrical Engineering* (4 credit hours)
  This is the freshman introductory course for students interested in electrical engineering. Students learn the design process, teamwork, ethics, and presentation skills.
- *ECE 102 Engineering Computation* (4 credit hours)
  Students learn how to apply computational software tools (MATLAB and Mathcad) to solve primarily electrical engineering problems.
- *ECE 103 Engineering Programming* (4 credit hours)
  Software design, algorithms, data structures, and documentation are covered using the C programming language.

In this new scheme, the original EAS course is split into two separate ones. ECE 101 spearheads the drive to give students early exposure to lab experiments and design projects, while ECE 102 carries over the engineering analysis principles and adds extensive MATLAB material. ECE 103 completely replaces the CS programming class and offers a focused approach for teaching programming to engineering (as opposed to computer science) students.

ECE 101 Exploring Electrical Engineering

ECE 101 is designed primarily for incoming freshman who are either planning or considering becoming ECE majors. The course outcomes include the ability to:

1. Do online research on areas of electrical engineering and present findings in a written summary.
2. Perform simple lab experiments to become familiar with lab equipment and components and to present the results in a lab report.
3. Complete a project involving both design and technical elements and to present the project in an oral presentation and a written report.
4. Work together as a team to successfully complete the project.

Class time is split between a classroom and a lab. The classroom time is spent on a variety of activities:

- Outside speakers from ECE faculty and local companies are invited to talk about different areas of electrical and computer engineering.
- The course instructor introduces some basic ECE content such as simple resistive circuits and logic gates.
- There are class discussions and group activities on team building, ethics, and technical communication.
- Students work on their team projects.

Homework is assigned weekly and usually consists of researching and writing a few paragraphs on an area or application of electrical or computer engineering. In the lab, students do simple experiments designed to introduce them to the lab equipment, components, software, and experimental techniques they will see in their later courses.

The main part of the course is a quarter-long, hands-on team project. Two projects have been offered so far:

- Build a Rube Goldberg contraption to accomplish a simple task by the most complicated means possible. The constraint is that at least three electrical engineering elements, such as sensors, motors, switches, LEDs, and electromagnets, must be included.
- Take engineering to middle school. Create: (i) a short and lively presentation on ECE for a middle school class and (ii) a hands-on activity to do with the class.

For either project, a written report and a presentation/demonstration are required. Students are randomly assigned to teams of four or five. Teams are allowed to choose which project to do. So far the great majority have chosen the Rube Goldberg machine. Students design and construct table-top setups that employ numerous steps to accomplish a simple task such as turning on a light bulb. Steps are traditionally mechanical, for example, balls rolling down ramps and knocking things over. For the electrical components, teams have the use of a lab kit for the quarter, but they can buy or scrounge additional parts from home. Students have shown great creativity in using objects from thrift stores, kitchens, and toy chests.

ECE 102 Engineering Computation

This is the follow-on course to ECE 101, and it explores engineering problem solving in detail. During the course development phase, questions were asked on how to structure it – should it be lecture only, lab-based only, or a combination of the two? What software package could serve both as a numeric calculation tool and as a way to introduce programming concepts? In the end, we decided that a mix of lecture material, limited lab exercises, and MATLAB[6] would best meet the desired course outcomes, namely the ability to:

1. Analyze and find solutions to engineering problems by applying the engineering method.
2. Use software tools to process data, perform calculations, and create graphs.
3. Analyze DC circuits using Ohm's law, node-voltage, and current-mesh methods.
4. Develop algorithm design skills for writing MATLAB scripts to solve simple engineering problems.
5. Investigate data acquisition and control via MATLAB programming.
6. Document a design project in a technical report.

While topics such as problem solving techniques, error analysis, and technical presentation are retained from its EAS predecessor, most non-electrical engineering subjects have been removed. This leaves more time available to discuss circuit components, DC circuit analysis, data acquisition procedures, and computation tools. The course expands the MATLAB coverage to over four weeks, along with adding supplemental Mathcad and Excel tutorials. Emphasis is placed on learning how to use software to calculate numeric solutions and generate correctly formatted graphs. As the course's primary computation tool, MATLAB was selected for several reasons: a) it is widely used in engineering classes and in industry, b) students benefit from learning to use math software early in the problem solving process, c) it serves as a "gentle" introduction to programming and acts as a bridge to the more advanced studies in ECE 103.

As a final project, students are required to write a MATLAB program that can control external sensors and actuators. This helps to reinforce essential programming concepts in the minds of students as they interface with "real-world" hardware to create a useful or entertaining electrical circuit. The interfacing is done via a LabJack[7] adapter, which contains both analog and digital I/O modules. It has a MATLAB-compatible application programming interface (API). The external circuit is attached to the LabJack, which is then connected to the host computer by a USB cable. Initial training is accomplished by several in-class exercises, in which the entire lecture period is spent experimenting with these adapters in the computer lab.

For the month-long project, students are organized into groups of two and are provided with specifications, circuit components, and an assembly schematic. Next, they design, code, and debug a program to interface the LabJack with the attached circuit. Finally, each group writes a report that thoroughly documents the design and testing process. To minimize the construction difficulty and cost, the hardware is purposely kept simple. The most recent projects have included:

- 3-bit binary up/down counter using individual LEDs for the display and a push-button switch to change the count direction
- 7-segment LED that displays various animated patterns with push-button switching

ECE 103 Engineering Programming

ECE 103 completes the sequence by providing a formal introduction to computer programming. Unlike its computer science counterpart, ECE 103 takes a more pragmatic approach to teaching programming by emphasizing engineering problem solving and spending less time on theoretical exposition. The stated course outcomes are for students to:

1. Apply the software design process to write programs that solve complex problems in engineering.
2. Employ variables, operators, built-in functions, and I/O statements to create simple programs.
3. Utilize selection and loop statements for decision-making and iteration.
4. Apply modular software principles by dividing programs into logical blocks.
5. Use arrays, pointers, and structs to develop more sophisticated data structures.
6. Control sensors and actuators by writing C programs to interface with LabJack adapters.
7. Document the project specifications, design, and source code.

Algorithm design and the software development process are discussed in detail. The chosen language is ANSI C89 because it is still widely used in industry[8] and provides a foundation for learning related languages such as C++. In addition, our department still requires C for some upper-division courses like digital signal processing and embedded systems. Since ECE 102 uses MATLAB to introduce fundamental programming concepts, more time is available in ECE 103 to concentrate on advanced subjects, such as pointers and linked lists. The transition from MATLAB to C is made easier by the similarities in both syntax and naming of common functions.

Building on their earlier MATLAB experience, students are exposed to a more rigorous software development process and to more complex programming tasks. The final project once again uses the LabJack, which also supports a C-compatible API. While the homework assignments use GNU C as the compiler of choice, the project utilizes the Microsoft Visual C development environment. Groups of two students are given a variety of projects from which to choose. Past projects have included:

- DC motor applications using a motor controller chip and pulse width modulation
  - Students used the motors in a variety of ways, such as powering a scale-model car and turning the drawing knobs on an "Etch A Sketch" toy.

- Digital timer using dual 7-segment LEDs
  - Due to limited I/O lines on the LabJack, students had to design their programs to multiplex the signals to the LEDs fast enough to minimize flicker.
- Morse code transmitter and receiver using an infrared emitter / detector pair
  - The transmitter converted typed-in English text to Morse code, which was sent over a modulated light beam to the receiver, decoded, and re-displayed as text.

4. Assessment and Evaluation Results

Our department uses a web-based tool[9] for direct assessment of student performance. It is used for evaluation of course outcomes as well as program outcomes but we will concentrate on the former. The basic idea is to use mapping of individual assignments in the course into course outcomes. For example, different questions on mid-term exam could be used to assess different course outcomes and would be mapped accordingly. Numerical scores on each such assignment are weighted by the contribution of that assignment to the overall course grade. An example of a spreadsheet with this information is given in Figure 1. Finally, for each course outcome a distribution of grades is generated based on all of the assignments used to assess it.

Grade Scale	A	90	B	80	C	65	D	50

	Homework				Quiz #1				Quiz #2			Quiz #3				Project		
Rubrics	**HW1**	**HW2**	**HW3**	**HW4**	**Q1.1**	**Q1.2**	**Q1.3**	**Q1.4**	**Q2.1**	**Q2.2**	**Q2.3**	**Q3.1**	**Q3.2**	**Q3.3**	**Q3.4**	**TECH**	**WR**	**Totals**
Max Points	15.0	15.0	6.7	13.3	2.2	3.3	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	19.2	5.8	***100.0***
Outcome #	1	2	3	4	2	1	1	2	3	3	3	4	4	4	4	5	6	
Student #1	14.3	13.5	6.7	12.8	1.1	1.1	2.2	0.9	2.2	2.2	2.2	2.2	2.2	2.2	2.0	19.2	5.0	91.8
Student #2	10.2	11.3	3.7	14.3	1.7	0.0	0.7	0.9	0.0	0.7	1.5	1.1	1.1	2.2	0.2	18.5	5.0	73.0
Student #3	14.0	14.2	5.8	13.3	1.7	1.1	2.2	2.2	2.2	2.2	2.2	2.0	2.2	2.2	2.0	17.3	4.6	91.2
Student #4	12.0	10.8	5.2	11.8	1.5	0.2	2.0	1.7	0.2	0.9	0.9	2.2	2.2	2.0	1.1	18.5	5.0	78.1
Student #5	13.7	12.7	5.3	11.7	1.5	0.2	1.7	0.9	0.2	1.7	1.3	1.5	1.1	1.7	0.0	15.4	5.4	76.1

Figure 1: Sample assessment data for ECE 102

It should be remembered, however, that all assessment tools are relatively coarse so we look for relatively large changes and trends, not specific, fine-grained distributions.

<u>*ECE 101 Exploring Electrical Engineering*</u>

The assessment in this class is based on homework, lab reports and the project. However, only the project is actually graded; the homework and lab reports are simply credit for completion. The intent of this class is to encourage a positive experience in electrical engineering without an emphasis on grades. Because of the number of class activities, attendance is also counted as part of the grade.

Not surprisingly, grades in the class are quite high, with over 90% of the class receiving A or B grades. Results are similar in 2009 and 2010, with enrollments of approximately 60 students per year. One more lab experiment, an introduction to LTSpice and MATLAB, was added in 2010 which not all students were able to complete, causing a decrease in that assessment category.

Students did not find the homework difficult, but, perhaps because it was not graded, students were sometimes lax about turning it in.

The projects were for the most part very successful, with over 90% of the students in 2010 receiving an A on their reports and presentations. Students worked with little assistance from the instructor or TAs and showed great creativity and initiative. The Rube Goldberg projects were impressive, both in the number of mechanical steps and in the inclusion of electrical components beyond what students learned about in the class. The only problems arose in a few groups that had trouble with team dynamics. Students with conflicting work and school schedules had difficulty meeting outside of class, and some groups did not feel all members contributed equally.

Students complete a questionnaire at the end of the course which asks for what they learned, what they found that was helpful or not, and suggestions for improvement. The response is very positive with the project, speakers, and labs all being very well received.

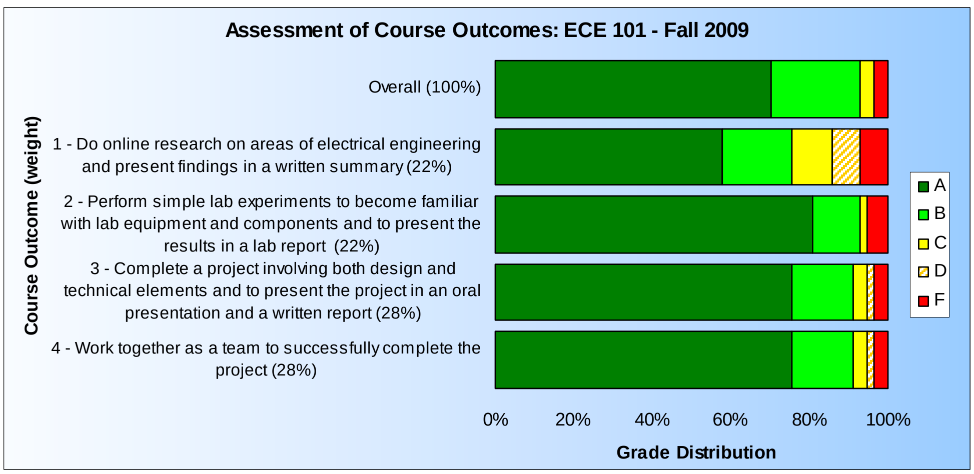

Figure 2: ECE 101 Fall 2009 (October – December) Results

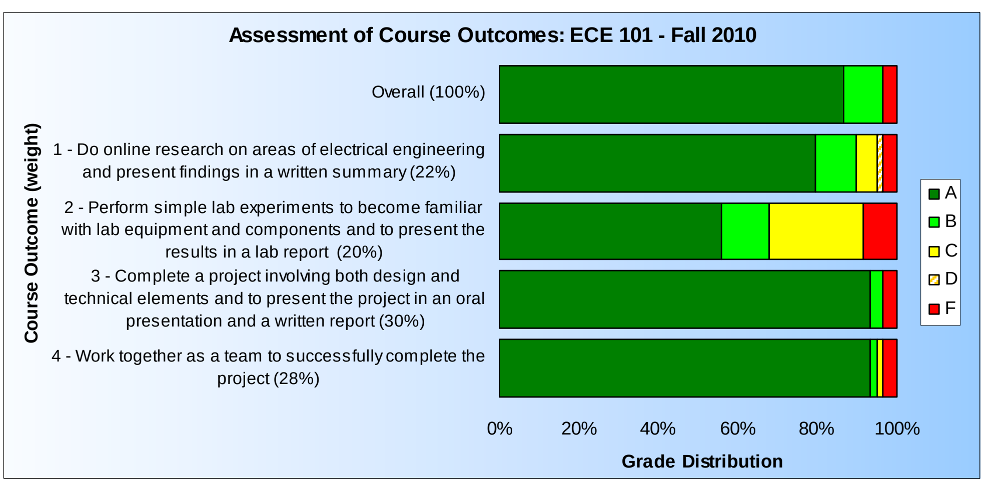

Figure 3: ECE 101 Fall 2010 (October – December) Results

There are questions of interest, not just for ECE 101 but for the entire sequence, related to retention and student performance in follow-on courses. We are developing appropriate metrics and applying them to the 1st and 2nd cohorts of students as they progress through their sophomore, junior, and senior years.

ECE 102 Engineering Computation

For ECE 102, assessment tools include homework assignments, quizzes, and the project. Homework consists of both manually calculated and computer-assisted solutions. Students are required to demonstrate their project circuit and MATLAB program to the instructor and submit a final report. Overall course grades are assigned on a straight percentage scale instead of on a curve.

The first deployment of the course took place in Winter 2010. With an enrollment of 65+ students, 90% of the overall outcome results were in the A, B, or C range. For several individual course outcomes, the combined number of A and B-level students exceeded 50% by reasonable margins (see Figure 4 - outcomes #2 through #6).

A few problem areas were evident. The number of A-level results for outcomes #1 (analysis) and #2 (software tools) was low at under 10%. While students generally understood the mechanics of analysis, they were inconsistent in applying the procedures in their solution attempts. This may be due in part to students not receiving enough homework practice before the instructional pace moved on to the next topic. Note that the assessment results did not include the effect of extra credit assignments, which would have improved the analysis ratings somewhat. A few students complained about the ordering of the topics, with MATLAB not being introduced soon enough for them to learn the software adequately.

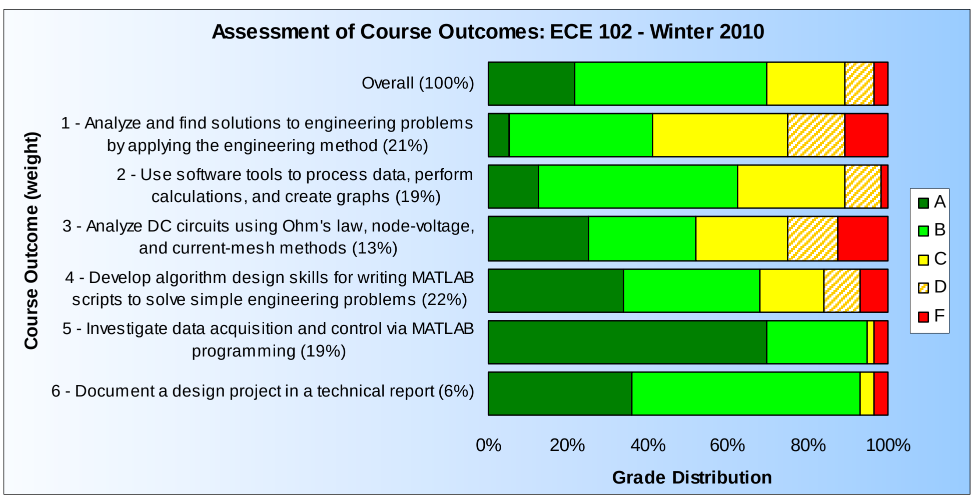

Figure 4: ECE 102 Winter 2010 (January – March) Results

The course was taught again during Summer 2010. With a much smaller enrollment of eleven students, students could be given more personalized help. Using the experience gained from the previous class, the schedule was re-arranged to make the topics flow more logically. MATLAB was introduced sooner and integrated more tightly with problem solving discussions and homework assignments. This resulted in noticeable improvement in key areas. The number of A-level grades increased to 50% for analysis skills (see Figure 5 - outcome #1).

Some deficits were present, especially for outcome #4 (algorithm design). This may be an artifact of a compressed schedule for the summer course (9 weeks versus 11 weeks). Also, students were asked to perform the project on an individual basis this time, which removed the benefits of team-based software development. Since we only have one year's worth of data it is difficult to draw any firm conclusions, but we will follow up this study with more detailed, multi-year longitudinal studies.

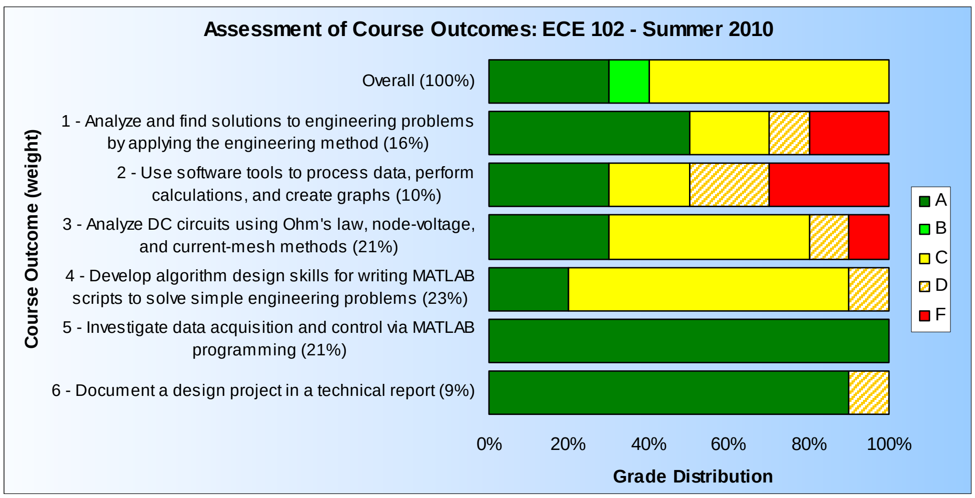

Figure 5: ECE 102 Summer 2010 (June – August) Results

The most surprising aspect of the course was the ability of nearly every group to complete the MATLAB-hardware interfacing project on time and to specifications. Even students who had little prior experience with programming and hardware were mostly able to meet the minimum standards. Some even added features for extra credit. Several students mentioned that while the project was at times difficult, the hardware aspects made the programming more enjoyable and gave them a better understanding of the code-test-debug development cycle when dealing with interfacing issues.

ECE 103 Engineering Programming

For ECE 103, assessment tools include homework assignments, quizzes, and the project. For each homework set, students are required to submit source code and a written report that discusses the design methodology. The final project requires a live demonstration and submitting of a comprehensive report. Overall course grades are assigned on a percentage scale.

In Spring 2010, the course debuted to an enrollment of 54+ students, with just under half having taken ECE 102 in the previous quarter. Those who had not worked with the LabJack before were given additional training to compensate. For all the outcomes, combined A and B-level scores met or exceeded the 50% mark (see Figure 6 - outcomes #1 through #7). For this course, both a teaching assistant and a tutor were available, so students had many resources for getting help.

Certain areas did not fully meet expectations. Only 10% of the students achieved A-level performance for outcome #1 (software design process), which partly measured how well students devised efficient algorithms and commented their programs. Students often relied on "brute-force" approaches instead of finding a more elegant solution, though this is not unusual for novice programmers and should improve with more experience. The quality of comments in the code was often inconsistent, especially for longer programs.

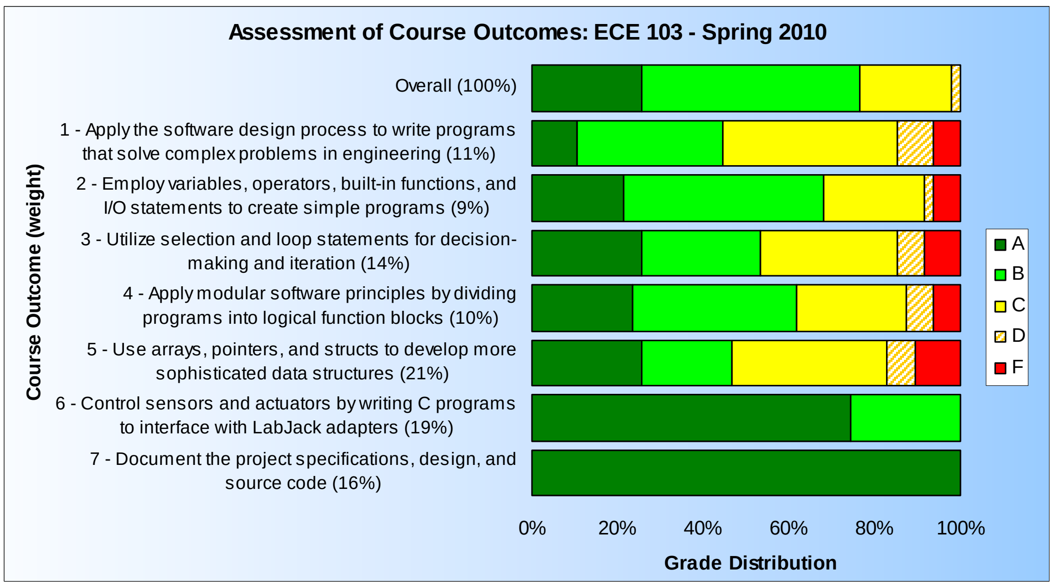

Figure 6: ECE 103 Spring 2010 (April – June) Results

The complexity of the projects in ECE 103 is greater in scope than in ECE 102. Circuit construction is more involved, and the programs require more sophisticated design work. However, students were still very successful in completing the projects, and the sense of accomplishment was reported as being greater due to the increased difficulty level. We have not yet confirmed if the MATLAB experience from ECE 102 had a significant effect on programming skills acquired in ECE 103.

5. Conclusions

Our objective was to address several programmatic issues: early introduction to design, enhanced programming skills, improved communication skills, and improved recruitment and retention. We accomplished this through a redesign of our full-year freshman electrical engineering course sequence, as follows:

- ECE 101 provides an engaging first-year experience while introducing design and team work through a fun lab project.

- ECE 102 provides a solid foundation in problem solving, writing, and programming with emphasis on MATLAB and simple hardware interfacing.
- ECE 103 expands the programming, design, and teamwork components by utilizing the C language in projects that re-utilize the LabJack adapter.

Based on our initial course assessments, we believe that students are achieving the curricular goals at the course level, but it is too early to assess the impact of this change on the whole program. Thus far, the integration of both MATLAB and C with hardware interfacing has proven to be successful as a tool for reinforcing programming fundamentals at the freshman level. Course syllabi, lab and project descriptions, and other assignments can be made available to colleagues who are interested in implementing similar changes.

6. Future work

After analyzing the assessment results, we are now exploring several avenues for improving course quality, raising student retention, and expanding the scope of our assessment methods, such as:

- Introducing new projects into ECE 101, in particular ones that engage the interests and needs of the local community.
- Finding ways to smooth the mental transition for students between the informal approach of ECE 101 and the more rigorous analytical techniques discussed in ECE 102.
- Performing longitudinal studies of student retention in ECE of those who took the freshman sequence compared to those who did not. In addition, we intend to track the retention of women and other under-represented groups.

Bibliography

[1] J. Spurlin, S.A. Rajala, and J.P. Lavelle, ed. "Designing Better Engineering Education Through Assessment: A Practical Resource for Faculty and Department Chairs on Using Assessment and ABET Criteria to Improve Student Learning," chapter 8, pp. 213-245, Stylus Publishing, 2008.

[2] D.W. Knight, L.E. Carlson, and J.F. Sullivan, "Improving Engineering Student Retention through Hands-On, Team-Based, First-Year Design Projects," ASEE 31st Intl Conf on Research in Engineering Education, 2007.

[3] M.J. Prince and R.M. Felder, "Inductive Teaching and Learning Methods: Definitions, Comparisons, and Research Bases," J. Engineering Education, vol. 95, no. 2, pp. 123-138, 2006.

[4] J. Richardson, et.al., "Freshman Design Projects in the Foundation Coalition," [Online]. Available: http://www.foundationcoalition.org/publications/journalpapers/AuthorHTML/R.htm

[5] J. Froyd, A. Srinivasa, D. Maxwell, A. Conkey, K. Shryock, "A Project-Based Approach To First-Year Engineering Curriculum Development," Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference , vol., no., pp.T3H, 19-22 Oct. 2005

[6] H. Diefes-Dux. (2004, May). *Purdue University Standardizes on MATLAB to Provide First-Year Students with Core Engineering Skills* [Online]. Available: http://www.mathworks.com/products/image/userstories.html?file=45529&title=Purdue%20University%20Standardizes%20on%20MATLAB%20to%20Provide%20First-Year%20Students%20with%20Core%20Engineering%20Skills

[7] LabJack Corporation Homepage [Online]. Available: http://labjack.com

[8] n.a. (2011). *TIOBE Programming Community Index* [Online]. Available: http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

[9] P. Van Halen. (2008). *Web-based quantitative analysis and reporting of program outcome coverage and student performance* presented at Best Assessment Processes X. [Online]. Available: http://abet.cecs.pdx.edu/ece2/abet/bap_apr08.ppt