Instructional Use of Computers in a Hands-on Programming Course for First-Year Engineering Students

Dr. Janet K. Lumpp, University of Kentucky

Dr. Janet Lumpp is a professor in the Electrical and Computer Engineering Department at the University of Kentucky and Director of the First-Year Engineering Program in the UK College of Engineering.

Dr. Jennifer Lovely, University of Kentucky

Dr. Jennifer Lovely is a Lecturer for the First Year Engineering Program at the University of Kentucky from the Department of Chemical and Materials Engineering. She has two undergraduate degrees: Spanish and Biosystems & Agricultural Engineering and her master's degree is in Biosystems & Agricultural Engineering, all from the University of Kentucky. Her PhD is in Grain Science from Kansas State University. Her research interests are related to food security and reducing postharvest loss. She's lived in Venezuela, Italy, France, and Brazil and encourages engineering students to study abroad.

Laura Marie Letellier P.E., University of Kentucky

Faculty lecturer in the First Year Engineering program at the University of Kentucky.

Dr. David L. Silverstein P.E., University of Kentucky

David L. Silverstein is a Professor of Chemical Engineering at the University of Kentucky. He is also the Director of the College of Engineering's Extended Campus Programs in Paducah, Kentucky, where he has taught for 19 years. His PhD and MS studies in ChE were completed at Vanderbilt University, and his BSChE at the University of Alabama. Silverstein's research interests include conceptual learning tools and training, and he has particular interests in faculty development. He is the recipient of several ASEE awards, including the Fahein award for young faculty teaching and educational scholarship, the Corcoran award for best article in the journal Chemical Engineering Education (twice), and the Martin award for best paper in the ChE Division at the ASEE Annual Meeting.

Dr. Derek Lynn Englert, University of Kentucky

Dr. John R. Baker P.E., University of Kentucky

John R. Baker is an Associate Professor of Mechanical Engineering at the University of Kentucky Extended Campus Program in Paducah, KY. He received his B.S., M.S., and Ph.D. in Mechanical Engineering from the University of Kentucky in Lexington, KY. After obtaining his B.S., he spent three years working in the Plastics Division of Eastman Chemical Products, Inc. He entered his current position in July 2000.

Dr. Neil Moore, University of Kentucky

Neil Moore is an assistant professor (special title series) in Computer Science at the University of Kentucky, where he teaches in the First-Year Engineering Program.

Dr. John F. Maddox, University of Kentucky

Dr. John F. Maddox is an Assistant Professor of Mechanical Engineering at the University of Kentucky, Paducah Campus. He received his Ph.D. in mechanical engineering from Auburn University in 2015. His primary research areas are thermal management of high power electronics through jet impingement and thermal characterization of advanced materials used in aerospace and electronics cooling applications.

Dr. Julie Gordon Whitney, University of Kentucky

Julie G. Whitney, Ph.D. received her bachelor's in mechanical engineering from Purdue University, West Lafayette, IN in 1982, her masters in industrial professional technology from Indiana State University,

Terre Haute, IN in 1986 and her Ph.D in mechanical engineering from the University of Cincinnati, Cincinnati, OH in 1992.

From 1979 to 1982 she was a student and then development engineer at General Motors, then interspersed employment, family and degrees between then and 1992 working for TRW, Northern Kentucky University and leading an NSF grant to help young people get started in STEM careers. In 1992 she joined Texas Instruments in gas sensor design and in 1997 she joined Lexmark, Inc. After retiring from Lexmark in 2016 she joined the University of Kentucky in Lexington, KY where she is now a lecturer.

Mr. Doug Klein, University of Kentucky
Dr. Whitney C. Blackburn-Lynch, University of Kentucky

Instructional Use of Computers in a Hands-On Programming Course for First-Year Engineering Students

Abstract

This Work in Progress paper describes the instructional use of computers in first-year engineering courses at the University of Kentucky (UK), a large public land grant institution which is the primary source of engineering graduates in the state. The College of Engineering (COE) at UK has transitioned to a common first-year experience from previously separate departmental introductory courses paired with one of two required computer science courses. The goals for the new curriculum are to increase retention, standardize the number of credits in each degree program and allow students to explore before committing to a specific plan of study. For the new computer programming course, the content pairs the sensors and actuators commonly utilized by a variety of engineers with the programming skills needed to collect and interpret data. These skills carry over to the team design project that involves programming, sensors, actuators, construction and testing of a complete system. The programming skills carried into the sophomore level courses have been well received by some departments and less than satisfactory to other departments which has led to more deliberate independent programming assignments. In this paper, we describe the implementation of in a new course sequence for ~900 students per year and preliminary results from course evaluations across the engineering curricula. While the program has only been in place 3 years, the initial trends show the program is correlated to improving retention, teaching useful programming skills and improving a sense of belonging to engineering which have been shown to correlate to improved graduation rates. Further analysis is needed to compare responses across each engineering and computer science discipline.

I. Introduction

Retention in engineering programs has become an increasingly important goal at universities across the nation for a variety of reasons. According to the National Center for Education Statistics, efforts to increase retention nation-wide have resulted in a modest upward trend in first-year retention rates at public colleges and universities as shown in Figure 1. [1]
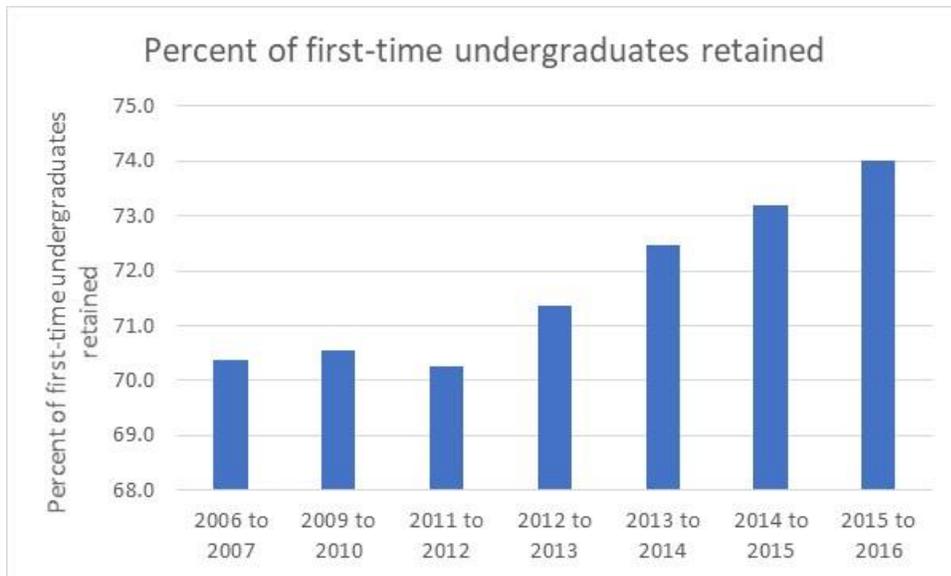
Figure 1. The average of retention rates for first-time undergraduate students returning for the second fall in 4-year degree granting public institutions by cohort year. [1]

Retention rates are important as they are a key metric of the amount of resources wasted, both financially and human, by the students and the institutions that attempted to support them. Students, who attempt a degree but drop out, have no saleable skills to show for the enormous effort put forth and the funds expended. Similarly, institutions do not have graduation rates that reflect the expenditure in operating and capital expenses they have made to teach these students. From an institution and at a personal level to the students and their families, those who attempt a four-year degree program but are unsuccessful have expended their limited resources.

Additionally, as technology becomes a larger portion of our modern life, the need for individuals with technical skills grows. Therefore, the inability to retain students who could become degreed, practicing engineers has a detrimental impact on society as a whole. Retention of students who are interested in a career in engineering is therefore a way to both reduce expenditure of resources and while also increasing the supply of skilled workers for tomorrow's job market.

II. The Challenge of Increasing Retention

Students leave the difficult path of an engineering education for a variety of reasons. Studies have shown that the primary reason is not due to the difficulty of the material per se, but rather the toll of emotional and physical stress of the college environment which includes coursework difficulty. [2] Improving retention does not necessarily mean a reduction in the quality of work of the resulting graduate, because it is possible to address this root cause of emotional and physical stress without sacrificing curriculum rigor.

The difficulty in addressing this issue is complicated by the physical and financial constraints on modern institutions, which face both rising costs and lower revenues from traditional sources, such as government entities. The cost of the program per student needs to be balanced between the cost to the individual and what is provided by the university in return. Economy of scale allows the university to purchase materials in bulk, negotiate software site licenses, staff technology resource centers and deliver course content through learning management systems in ways that would be cost prohibitive to most students. Education costs also include materials like laptops, campus-wide technology infrastructure, classroom technology and instructor time spent per student. All these factors were considered when developing a new First-Year Engineering curriculum to be taught in reconfigurable classrooms located in a Living Learning Community residence hall.

Ossman and Bucks have identified multiple examples of institutions where the incorporation of engineering projects involving hands-on work has improved student outcomes [3]. This work has generally utilized a software development tool such as MATLAB and a hardware platform. The use of hands-on activities utilizing MATLAB and Arduino to improve student performance has also been demonstrated by Caglayan et al. in a first year engineering summer camp for incoming students [4]

MATLAB and hardware platforms such as the Arduino Uno require use of a personal computer or laptop to program. To adopt a similar teaching approach therefore mandates a decision as to how best to ensure that students have the necessary laptops or access to personal computers. It also mandates cooperation with the Living Learning Community (LLC) who shares resources with the FYE program.

The LLC is a central recruiting and retention initiative of the College of Engineering and has moved to larger residence halls as the program has grown. In the newly constructed LLC, two classrooms were integrated into the facility for use by both the FYE courses and LLC activities. Students are provided multiple opportunities to interact with other residents, classmates, peer mentors, resident directors and faculty to create a sense of community, prevent isolation and ensure that students are aware of all the resources provided by the university.

Regarding the instructional use of computers in FYE, the driving factor for the choice of technology is that space in the residence hall is shared by the FYE and the LLC.   Installing desktop computers would have been incompatible with LLC activities and expensive to maintain, therefore, the choice was made to institute a laptop requirement for incoming engineering students. The more the institution takes ownership of the acceptable laptop models, the more it can control the performance, compatibility, and reduce the support staff needed to keep systems functional. Conversely, the more the students have freedom to select their laptops the more they can meet their individual financial constraints while also feeling more in-control of

their education. Students take pride in their laptops and personalize them with colorful stickers, screen savers and wall paper. Issuing students laptops was cost prohibitive for their institution and the infrastructure already existed for technical support of a wide variety of laptop platforms. For the purposes of the program, specifying a minimum performance requirement and allowing students to source their own laptop was the solution.

When considering ways to reduce stress on incoming students, FYE was designed to allow more flexibility in the first year to select an appropriate major. However, providing this time for students comes with complexity for the FYE faculty of then ensuring that the material taught in the first year is applicable and adequate for all the majors that the program is designed to feed into. Another complicating factor is that resources committed by the university for technical equipment have a limited useful life due to changes in technology as well as wear and tear of the students' laptops while in use, and these investments need to be constantly renewed. Finally the large number of students who enter today's engineering programs at many institutions provides its own logistical barrier to retention efforts. Developing laboratory experiences that promote personal discovery are made more difficult the larger the number of students that must have access.

III. Creating a First-Year Computer Programming Course

Students entering a large engineering program bring a wide variety of background skills. While the ability to understand how computers and computer programming works, and what these tools can enable is considered to be a basic skill of the modern engineer there is a wide range of how computer-ready students are as well as the computer literacy needs of the different engineering majors that will inherit them from a first-year program. Improved retention through flexibility in engineering discipline selection in the first year is effective, but comes with the complication of compatibility of first year content with discipline specialties that follow.

Through careful discussion it was concluded that a 2 credit-hour EGR 102 Fundamentals of Engineering Computing would be included in the first-year curriculum. This was a compromise made by all disciplines in the College of Engineering. There was concern that a standard computer programming course would be a "turn-off" for many students who prefer hands-on design and discovery. Engineering disciplines such as Civil Engineering and Biosystems Engineering were concerned that students would leave the program for lack of enjoyment and motivation before they ever reached their discipline specific course work. Alternatively Computer Science and Computer Engineering disciplines were concerned that a weak course would not prepare students adequately for sophomore level programming instruction, and these students would leave the program in frustration later resulting in even greater resource waste. Addressing these concerns, the Fundamentals of Computing course was designed to be a more active hands-on course where programs would be written in MATLAB and these programs

would not only be tied to interdisciplinary engineering problems, but "brought to life" by creating working systems using Arduino microprocessors, simple electronics kits, and a selection of sensors and actuators. Students are encouraged to work together with their neighbors and students who finish in-class activities quickly are asked to help others around their table to further practice with the content. McNeil et al observed that team activities in a first-year computer course improved student perceptions of the value of computer programming and reduced the time spent on programming assignments. [5]

III.A. Laptops and Microprocessor Kits

Both laptops and microprocessor kits represent a significant investment in technology for students. The laptop is both expensive and has a limited lifespan due to rapid technological changes or the rigors of student life. Institutions have the option of requiring students to provide laptops of their own that meet minimal requirements, specifying a particular laptop and installed software or providing computers in computer labs or issuing laptops or tablets to students. Each of these options has its benefits and detriments. At the start of FYE, we selected the first option of setting specifications for personal laptops. Most students have been able to comply with the requirement and we have temporary loaner laptops to cover short term problems with the personal laptops. We are reconsidering the second option of specifying one of a set of prescribed systems in order to reduce the technical support workload required to help students maintain the wide range of machines they bring to campus.

Student sourced laptops did have several unforeseen issues that any program considering this approach should be made aware of. While the performance requirement for the College of Engineering is that every laptop be capable of running the Microsoft Office suite of products and run advanced software, for example CAD packages, that require a Microsoft Windows operating system, that requirement has been difficult to enforce. As a result instructors have had to change their strategy for software to accommodate the fact that about 10% of the students will have issues with any required software. For the First-Year Program selection of MATLAB and the add-on software package to run Arduino (a microprocessor) from MATLAB has proven extremely robust against the large variety of laptops on which it must run. CAD software strategy has been to move to packages available in the cloud, so that compatibility on the individual laptop is not an issue in the same way that hosted learning management software such as Canvas and Blackboard exist mostly independent of the computer hardware that the student accesses them from.
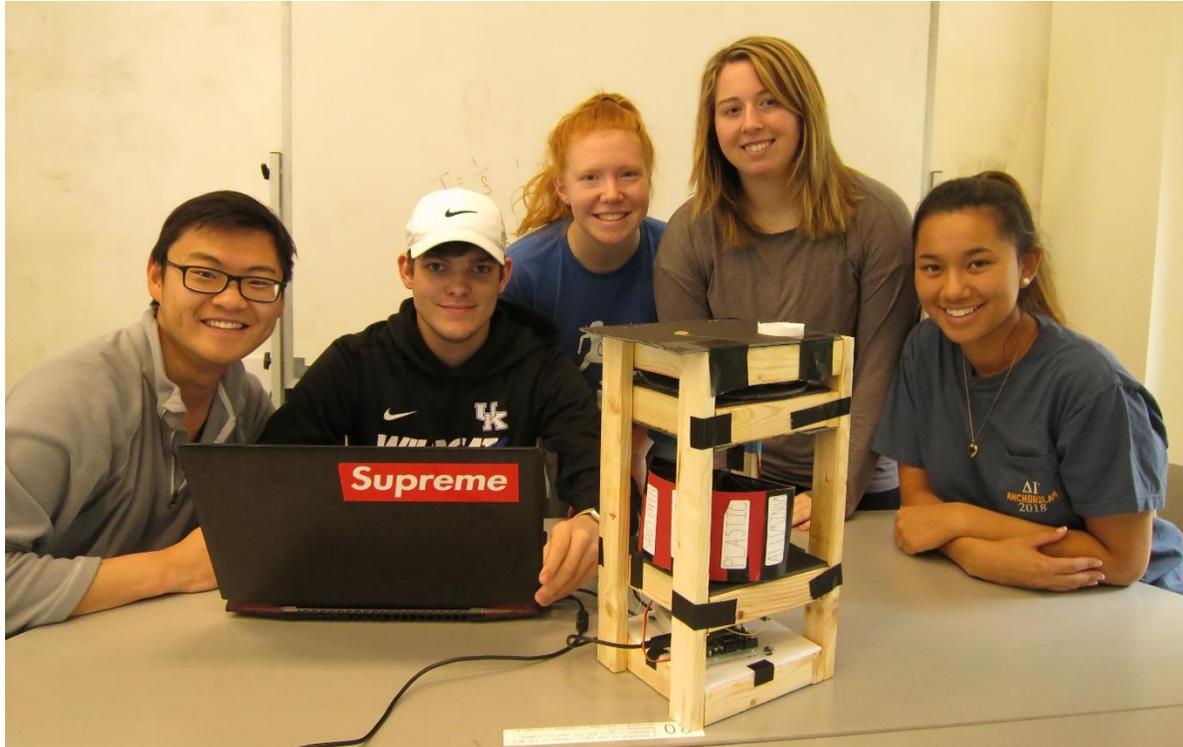
Figure 2: Students use their own laptops combined with their microprocessor kits (see Arudino board and circuit at the base level of this design project) which is supplied by the university, in several of their First-Year Engineering classes, including Fundamentals of Computing.

Another set of issues that arise within a student-provided laptop requirement is the potential for laptop failure or a student's inability to purchase their laptop prior to the start of classes. For these students the college provides loaner laptops, and specifically in the Fundamentals of Computing and other First-Year Engineering classes, a cabinet of charged laptops is always available. Students trade their student ID card for use of the laptop during class and the laptop is set up so that any new files are erased when the laptop is closed. This means that the student who shows up in class without a working laptop, or one that due to compatibility issues cannot be made to work, does not miss any instruction time. The loan for the class is made on the spot. This ability to get a loaner laptop has the effect of also reducing stress on students, especially those with limited financial resources.

The opposite strategy was used for the sourcing of the Arduino microcontrollers and associated electronics kits. These kits were selected by the First-Year Engineering team and represent an investment by the university. These kits are issued to students at the start of their first-year classes, specifically the Fundamentals of Computing class and are not collected until the completion of the final first-year course, Engineering Exploration II. Figure 2 shows a team proudly displaying their final design project including an Arduino controlled by one of their laptops. The kits are then refurbished and reissued to students coming in the next fall. Failures and changes in compatibility are the responsibility of the institution, rather than the individual

student. Kits are predictable and compatible with software and activities, and purchasing in quantity allowed the university to secure better pricing than students could individually achieve. It is interesting that the students do not emotionally take possession of their kit the way they do their laptop. Kits are frequently left in the classroom by careless students, however, the same is never true of their laptops. All kits are provided in a durable plastic box and a sticker on the outside of the box has the student's university email for identification and to re-unite student and misplaced kit. Interestingly theft of kits has not been an issue even though students are aware of their monetary value.

III.B. Hands-on Activities

The laptop and microprocessor kit become the base on which the hands-on programming activities are built. Careful attention is paid to finding engineering applications for simple circuit structures. For example a voltage divider circuit creates a sensor out of a photoresistor. Breaking a beam of light is used as a common sensing means for monitoring many manufacturing processes to determine the presence of an object. Students build a "food line monitor" that counts the number of cans passing a given point. The computer logic of while loops and if/else statements fit nicely into this format. Switches make good logic components and early in the semester students build their own float switches to monitor a fluid level and potentially alarm (an LED lights) or turn on a pump (start a small servo motor).

The properties of circuits themselves provide useful technical content appropriate for all engineering students as well as giving additional programming learning opportunities. Early programming activities calculate total resistance in a circuit and alert of too much current would flow. Programs also check the presence of voltage in certain areas as input, and scale voltage output to other areas. The course structure is generally one program without a circuit component on one day to introduce a programming concept, and then on the second class meeting that concept is used in a circuit that actually performs a function that students can see. By keeping the activity prompts tied to understandable real-world engineering problems, the value of the tool is easily understood by students.

The course design also tries to introduce students to some of the more cutting edge technologies to allow them to feel that their efforts have current relevance. Students discuss data analysis and manipulate data from real data sets such as water level monitoring from local streams or food safety data for baby food. They also build a reflective light sensor, gather data and use a simple machine learning tool to make and train a roughness sensor shown in Figure 3. From these modest activities they are introduced to the fields of Artificial intelligence and Big Data.

Fundamentals of Computing Student Made Roughness Sensor
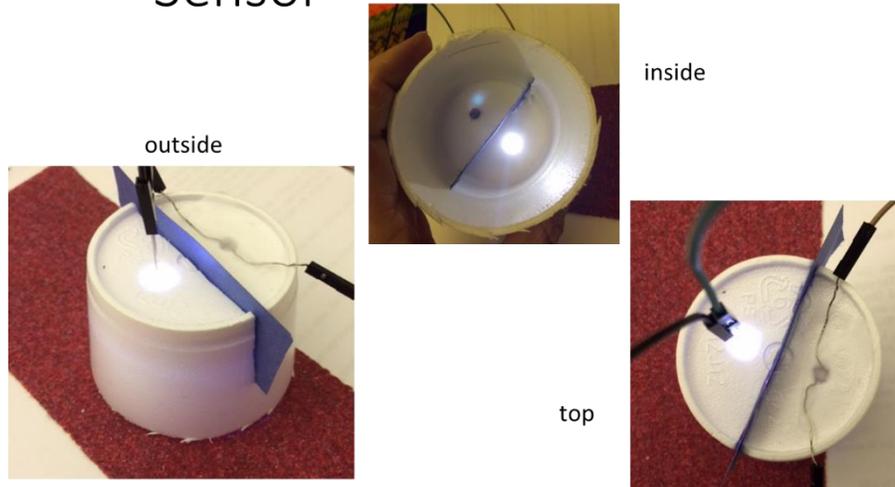

inside

outside

top

Figure 3: Example of a sensor built by students in Fundamentals of Computing. The sensor consists of a photoresistor in a voltage divider circuit and an LED. The program collects data from materials of different roughness and uses it to train a simple decision tree machine learning algorithm. Students then implement the algorithm to make a classifier out of their sensor.

III.C. Logistical Considerations

As mentioned in the introduction, one of the challenges facing any organization wishing to do hands-on learning for large numbers of students is the logistics of getting what ideally would be individual instruction out to so many. To that end several strategies have been employed in the Fundamentals of Computing class that have met with moderate to high success.

Small class sizes is the goal of any program trying to give the best quality instruction. However the resource cost of small classes is very high. A compromise used by the Fundamentals of Computing course was a moderately large class, generally capped at just over 70 students, but staffed by an instructor plus 3 or 4 teaching assistants. This staffing allowed for an economical way to get individual help to students in the classroom setting to debug circuits or code without the frustration or stress of excessive wait times. Additionally, the First-Year Engineering team created "open lab" times, also staffed by teaching assistants, where students from any of the first-year classes can come get help.

Software support help is always directed to the university technology support hub, which is staffed to service and support student software and hardware needs. Relying on university resources makes the wide variety of laptop platforms a workable model for FYE classes.

Grading is another issue with large student populations. Teaching assistants can do some grading, but the Fundamentals of Computing course has more effectively tried to design assignments that "grade themselves". While for most institutions this would be programs that check the inputs and outputs of student programs, for this course reducing grading load means designing activities (programs and circuits) with an easily identified "success" criteria. Students or student teams demonstrate function for a check-off in many situations, and that is then entered into the grading system.

Exam taking and grading has been computerized so that students use their laptop to securely take an exam that is automatically graded for speed and error reduction. Students take common hour evening exams on their laptops and each exam room has paper copies to accommodate students with computer issues. The paper copy option also alleviates stress among who students who are concerned that their laptop may not be reliable. The exam scores are consistent across paper and computer-based exam takers. Team work on small projects also reduces grading load as the number of items that must be reviewed is reduced by a factor of the number of students in the team. These activities also give students a chance to make friends with other engineering students, a factor that can improve retention as well.

IV. Results

The goal of this paper is to describe an FYE program that is successfully addressing the need of improved retention while maintaining student quality using a variety of strategies, and specifically focus on one core FYE course. The course combines student laptop use, microprocessor use, carefully designed activities, and course structure to teach a key technical skill, fundamentals of computer programming, to first-year engineering students. After three years of running this course for approximately 900 freshmen and transfer students per year, data shows not only improved retention for the College of Engineering including through the second year of coursework, but also consistently high rankings by students as to the perceived value of the course.

Several metrics are used to measure the success of the FYE program and the Fundamental of Computing course, specifically. Institutional retention data for the first-time freshmen in the College of Engineering are shown in Figure 4 starting with the Fall 2003 cohort up through Fall 2018 noting the new FYE curriculum was introduced with the Fall 2016 cohort. The retention rates for the first fall to spring have been relatively constant for the past ten years but the percent retained for their second and third fall semesters are starting to rise since the introduction of

FYE. Although the FYE program may not be the only factor influencing retention, the timing corresponds to the curricula changes across the College of Engineering. The third fall retention rate had begun to rise with the 2015 cohort before FYE and no significant impact has been seen yet on the fourth fall retention rate. We are also collecting course-specific data to determine which of the FYE courses help students succeed in upper level coursework.

Cohort Term	Cohort Size	Cohort Size with Exclusions	Retained in College % 1st Spr..	Retained in College % 2nd Fall	Retained in College % 3rd Fall	Retained in College % 4th Fall
Fall 2003	430	428	75.3%	57.7%	42.3%	0.0%
Fall 2004	419	417	69.9%	53.5%	38.7%	0.0%
Fall 2005	433	433	71.1%	56.4%	46.9%	44.8%
Fall 2006	399	398	67.4%	50.1%	42.9%	40.6%
Fall 2007	379	379	75.7%	60.7%	50.7%	47.8%
Fall 2008	425	424	89.4%	66.4%	49.2%	47.8%
Fall 2009	561	555	88.6%	68.8%	53.5%	48.3%
Fall 2010	644	640	87.6%	64.4%	50.9%	46.1%
Fall 2011	602	597	86.5%	67.8%	54.7%	50.5%
Fall 2012	713	707	87.8%	69.4%	56.7%	50.1%
Fall 2013	684	680	85.8%	68.3%	52.5%	49.9%
Fall 2014	775	771	86.8%	69.4%	53.9%	51.2%
Fall 2015	780	777	87.2%	67.9%	59.2%	52.3%
Fall 2016	789	789	87.6%	75.7%	62.1%	52.2%
Fall 2017	668	668	86.5%	75.0%	58.7%	
Fall 2018	753	752	88.3%	79.2%		

Figure 4. College of Engineering retention data for first-time freshmen showing increased retention after the introduction of the FYE Program in Fall 2016.

Every course is evaluated by students at the end of the semester using on-line surveys with standard and customizable items using either a 5-point Likert response scale or open response textboxes. The two items commonly used to compare courses and rate faculty are: "I consider this to be a quality course." and "The instructor provided quality teaching.". Table 1 summarizes the student responses for the large enrollment semesters of the Fundamentals of Computing course. The means are based on a 5-point scale from strongly disagree (1) to strongly agree (5) and the mean over all courses in the College of Engineering are provided for comparison. FYE Faculty are rated very highly by students for quality teaching in all the FYE courses and our scores exceed the average for all faculty in the College of Engineering. The course was launched at full capacity in Fall 2016 with only minor implementation issues and received positive but below average course quality scores across 16 sections. After some revisions during the off-sequence semester, the course has consistently received above average course quality scores in high and low enrollment semesters. The Fundamentals of Computing, in spite of being a rigorous course, receives high ratings from students as a course that has a perceived value to them.

Table 1. Mean course quality and instructor quality scores from end of semester evaluations for three high enrollment semesters of the Fundamentals of Computing course. The college-wide mean is from Fall 2018.

	Fall 2016	**Fall 2017**	**Fall 2018**	**College**
Invited	997	797	826	
Responded	746	516	540	
Response Ratio	74.82%	64.70%	65.40%	
Mean (Std. Dev.)				
Course Quality	3.35 (1.13)	4.2 (1.0)	4.1 (0.9)	4.0 (1.1)
Instructor Quality	4.31 (0.82)	4.5 (0.7)	4.4 (0.8)	4.2 (1.0)

Starting in Fall 2018, several items were added to the end of semester evaluations for more detailed assessment of the impact of FYE on first-year students and across each degree program. A 4-point Likert scale was used for these items to eliminate the "neither disagree or agree" option and encourage students to respond one way or the other. One set of questions was added to the FYE courses to determine whether the courses are having the intended impacts on choice of major, developing problem-solving skills, practicing teamwork, establishing a peer group and preparing for future coursework. Another set of questions was added to a select group of sophomore, junior and capstone design courses in each undergraduate degree program. These questions ask students in each major to reflect on whether the skills they learned in their first-year courses were applicable in the current courses in their major.

Table 2 compares first-year student responses to "This course helped me establish a peer group/I made friends in this course." with responses from sophomores, juniors and seniors to a similar statement, "I feel connected to the community in my department/college." First-year students responded in agreement that they established a peer group in their Fundamentals of Computing section. The junior level courses were not all offered in the fall semester which accounts for the lower number of responses. The seniors would have had department specific first-year courses before the launch of the combined FYE curriculum. It is possible that the sophomores feel somewhat less connected in their major than they did in the FYE program until they have taken several courses in their department. Nevertheless, all the cohorts agree that they have a sense of belonging to a community.

With regard to computer programming skills, the same groups of sophomores, juniors and seniors responded to "This course used programming and/or software skills that were introduced in the engineering coursework I took during my first year." The sophomores would have taken the Fundamentals of Computing course whereas the upper level students may have taken other introductory programming courses. Not all courses in every major require computer programming skills which may cause a student to disagree with the item. Table 3 summarizes the responses to computer programming item across multiple disciplines. Further analysis is

needed to compare the programming content across the courses and to determine the percentage of students in each cohort who have taken the new FYE programming course.

Table 2. Mean responses and response rates to items regarding community building in FYE and departmental courses in Fall 2018. Sophomore scores are from CE 211, CME 200, CS 215, EE 211, ME 205 and MSE 201. Junior scores are from CS 275, EE 461 and ME 325. Senior scores are from capstone design courses CE 429, CME 455, CS 499, EE 490, ME 411and MNG 591.

	FYE Students		
Invited	826		
Responded	540		
Response Ratio	65.38%		
"This course helped me establish a peer group/I made friends in this course." – Mean, 4-point scale	3.13		
	Sophomores	Juniors	Seniors
Invited	527	195	355
Responded	333	86	198
Response Ratio	63.19%	44.10%	55.77%
"I feel connected to the community in my department/college." – Mean, 4-point scale	2.83	2.80	3.06

Table 3. Mean responses and response rates to the item regarding computer programming in FYE and departmental courses in Fall 2018. Sophomore scores are from CE 211, CME 200, CS 215, EE 211, ME 205 and MSE 201. Junior scores are from CS 275, EE 461 and ME 325. Senior scores are from capstone design courses CE 429, CME 455, CS 499, EE 490, ME 411and MNG 591.

	Sophomores	Juniors	Seniors
Invited	527	195	355
Responded	333	86	198
Response Ratio	63.19%	44.10%	55.77%
"This course used programming and/or software skills that were introduced in the engineering coursework I took during my first year." – Mean, 4-point scale	2.18	2.02	2.49

FYE has worked closely with the Computer Science department to ensure that students are prepared with adequate programming skills to succeed in advanced CS coursework. In Fall 2018

we added more complex programming assignments to challenge students to develop individual skills and not rely on team members in class. We will measure the impact of the new assignments in future semesters. The preliminary responses from CS students are presented in Table 4 showing greater agreement from senior computer science students than other CS or College of Engineering students to the item about whether their capstone course used programming skills from their first-year coursework.

Table 4. Mean responses and response rates to the item regarding computer programming in CS 215, 375 and 499 in Fall 2018.

	Sophomores	Juniors	Seniors
Invited	105	64	63
Responded	38	26	22
Response Ratio	36.19%	40.63%	34.92%
"This course used programming and/or software skills that were introduced in the engineering coursework I took during my first year." – Mean, 4-point scale	2.48	2.13	2.71

V. Conclusions

The new FYE program at the University of Kentucky has begun to show increased retention for Fall 2016 first-time freshmen in their third fall and Fall 2017 first-time freshmen in their second fall. In order to launch the new FYE courses, the College of Engineering chose to provide microprocessor kits on loan to students for the year, obtain a university-wide site license for MATLAB and to require engineering students to bring a laptop that meets certain performance criteria. The University already provided technical support for all students and the College provides laptops for short-term use during FYE courses to eliminate down time due to malfunctions or forgotten charging cords. As a result, the Fundamentals of Computing course has been well-received by FYE students as shown by course quality scores at or above the average score for all College of Engineering courses.   While there are still non-FYE students in the pipeline, the program has captured end-of-semester course evaluation data on student perceptions of whether their first-year coursework prepared them for upper level classes in their majors. Preliminary analysis of the first data set shows a neutral response from sophomores, juniors and seniors regarding the utility of their first-year programming courses. Students from each stage of their academic careers agreed that they have developed a sense of community in the College of Engineering.

VI. Future Work

To make the Fundamentals of Computing course even better for students and the university, the team is constantly trying to improve the hands-on projects and programs to keep them relevant and addressing the needs of the downstream engineering disciplines. Working with the Computer Science school several programming assignments have been and are being modified to insure students who select computer science as a major have had the requisite experiences to ensure success. Simultaneously the fact that students already have microprocessor and programming experience can now be leveraged by several disciplines who had to teach those skills previously, and the team plans to work with those groups as the first student though the program reach those classes, to help instructors leverage student capabilities.

Additionally, we are trying to learn from this successful course more about what is causing it to be well-received by students so that those lessons can be applied to other retention efforts, including other First-Year Engineering courses. We will continue to analyze the added course evaluation questions across multiple semesters and begin to code the open responses where we expect to find themes that will inform further course revisions. As the FYE cohorts move through their curricula we will look for evidence of the impact first-year coursework will have on upper level student success.

VII. References

[1] National Center for Education Statistics, "Undergraduate Retention and Graduation Rates," May 2018. [Online]. Available: https://nces.ed.gov/programs/coe/indicator_ctr.asp. [Accessed 31 January 2019].

[2] Creditdonkey, "23 College Dropout Statistics That Will Surprise You," November 2017. [Online]. Available: https://www.creditdonkey.com/collegedropout-statistics.html. [Accessed 31 January 2019].

[3] K. A. Ossman and G. W. Bucks, "Adding Hardware Experiments to a First-Year Engineering Computing Course," in *Proceedings of the ASEE Annual Conference & Exposition*, Columbus, OH, 2017.

[4] O. Caglayan, S. Ande, E. Coronado, M. J. Martinez and S. J. Hondowski, "Summer Engineering Academy for First-year Students in STEM: Making the Transition to College Through Coding and Robotics," in *Proceedings of the ASEE Annual Conference & Exposition*, Salt Lake City, UT, 2018.

[5] J. C. McNeil, A. Thompson and N. Hawkins, "A Comparison of Students Learning Programming with Online Modules, Instruction and Team Activities," in *Proceedings of the ASEE Conference & Exposition*, Salt Lake City, UT, 2018.