

Using Student Owned Computing and Tablet PCs to teach Engineering Problem Solving using Excel and VBA to Textile Students

Jeffrey Allen Joines and Melissa A. Pasquinelli, N.C. State University, Raleigh, NC, USA

Abstract

Many engineering curriculum around the country are re-evaluating their introductory computer programming requirement. At our university, several departments have been changing from the traditional Java or C++ course to something more applicable to their discipline. This paper will address the development of a joint course that was taught separately in both across Industrial and Textile Engineering. Students from both departments were not using their programming knowledge in remaining courses or when they graduated. Furthermore, the introductory Java (C++) class was being taught as a service course to masses of students at one time in our university's theater. The students were not enthusiastic about the course and certainly not developing the computer modeling skills that we felt were necessary (i.e., given a problem can they develop a method for solving the problem). Therefore, we decided drop the introductory course and develop a new course in Excel and Visual Basic for Applications to better address the needs of our industry and faculty colleagues. This new course was not intended to duplicate the traditional computer science method of presenting programming. Our goal was to educate students to model problems relevant to our disciplines, solve these problems using modeling tools, and then analyze these solutions through decision support (i.e., become "power users" and not programmers). This paper will address the critical development of a series of "InClassLabs" and their impact on student learning and our two curricula. Many of our homework and cases studies come from industrial sponsor data and represent real cases. The paper will discuss the fundamental issues that lead to the development of this computer-intensive course. Of special interest is the classroom environment bolstered by the use of student owned computers, in-class teaching assistants and the use of Tablet PCs. Student evaluations are used to provide insight into the teaching strategies employed.

Introduction

The computing experience in engineering has, on one hand changed dramatically, and yet, on the other hand has not changed very much. Perhaps fifty percent of all engineering courses now employ some form of computing within the course (e.g., Excel, SAS, SolidWorks, Matlab, Maple, etc.). Computing is omnipresent within all engineering courses and most engineering tasks. However the introductory computing course taken by engineers has not fundamentally changed, except for the programming language. Engineering faculty who make a programming/computing assignment cannot depend on their students having the same programming language, let alone a uniform programming experience. To complicate matters further many of our students took their introductory programming course at regional community colleges. It has been observed that anyone foolish enough to make a programming assignment almost surely risks lowering their course evaluation. Borrowing an idiom "the more computing changes, the more the introductory programming course remains the same." Can the introductory programming course be changed to render it more beneficial both to the students and the faculty? Can we identify a rationale that allows us to create a more relevant computing experience? In this paper we make several observations regarding the computing experience for engineers and suggest that there are other approaches to this course.

Different engineering disciplines have different computing needs. Computer applications are typically discipline-specific. A consideration of course revision needs to recognize the needs of the discipline since civil engineers are not mechanical engineers who are not electrical engineers. In most of the service courses, generic academic problems are used. They do not relate to the student's discipline, and therefore motivation in this course is a major factor.

Computing for Industrial and Textile Engineering

Since our own expertise is in Industrial Engineering and Textile Engineering, we will focus on those similar (but different) disciplines. To keep this discussion generic, we will use the term "systems engineering" to encompass both Industrial and Textile Engineering. Systems engineers are modelers. In particular they model decision processes, namely the way decisions are made, especially operations management and design decisions. Invariably such decisions require decision support, namely the collection and compilation of information relevant to the decision. Thus computing for systems engineers needs to support the modeling of decision processes. More importantly, they need to model the problem by gathering all of the information needed (i.e., equations, data, etc.), solving the problem, and then presenting results in a manner that the decision can be made easily. One thing that is important which is often missed is that the solution needs to be appropriate to the assumptions of the problem. Furthermore the decision making process needs to be supported by data and facts. We call this "computer-based modeling." To summarize, the one introductory programming course in Java offered as a service by our computer science is not working for systems engineers for the following reasons: (1) It is taught in a large university theater without the use of computers except in a separate lab ; (2) The course is taught in the traditional manner of programming syntax (i.e., taught without regard to problem context); (3) The students often did not take this programming course at our University (i.e., no uniformity in student knowledge); (4) Lack of knowledge of Java programming by engineering faculty limits subsequent courses offering programming assignments for discipline context; (5) Lack of algorithmic and problem solving ability with computing; (6) Most of our graduates had no reason to use this programming course when they graduate in their jobs; and (7) Industry constituents want our systems engineers to be able to tackle problems using appropriate computing.

Designing a Computer-Based Modeling Course

Realizing that the standard introductory programming course no longer appropriately complements the education of systems engineers, it is natural to ask what kind of course is better suited. First, it is important to consider the computational nature of the discipline. There are several "fundamental" subjects. These include the methodological areas of statistics, optimization, simulation, polymer chemistry (in textiles engineering), and stochastic processes, as well as the informational elements of database analysis and design. Applications then extend to engineering economy and financial modeling, work design and ergonomics, quality monitoring and control, logistics, and production systems, as well as polymer chemistry, textile processes, and fiber physics. A wide variety of software are used including SAS, Lingo/Lindo, Arena, Excel, Access, Matlab, Maple, AutoCad, ProEngineer and many others. Altogether these contain wide-ranging computing features including graphical interfaces, modeling, expressions and equations, algorithms, data structures, and numerical computation.

Looking for the Computing Essence

It is important that the introductory computing course in an engineering discipline represent the "kind" of computing that the discipline requires. Also, we do not want to "throw the baby out with the bath water", namely any introductory course needs to incorporate programming. Programming remains

the most fundamental computing tool and is critical to understanding “modeling” as incorporated in application languages. For example, Arena simulation uses “point, drag and drop” methodology for simulation modeling, yet specifications of the model components relies on Arena “expressions” that have their origins in programming. To understand how to employ Arena fully means that an understanding of programming is essential. The same is generally true of most engineering computing applications. Also, often these applications may not be able to model the particular problem and further enhancements are needed which can only be accomplished by programming.

Any introductory computing course in systems engineering needs to have both a modeling and a programming component. The modeling needs to reflect the way the discipline approaches problem-solving while the programming needs to support problem-solving. Therefore, we do not want to simply substitute a more appropriate language for Java and then teach it in the same manner. Therefore, we are arguing that we need develop a problem based approach that requires computing to solve these problem (i.e., teaching modeling and programming within the context of discipline specific problems). This addresses why the current programming course was not appropriate. The next section will describe our approach and the success of this approach.

Choosing a Tool/Programming Environment

Perhaps the most accessible and available modeling tool is Microsoft Excel. The spreadsheet software has evolved for over almost 25 years of development, starting from a simple matrix of values. It is a visual display of values that is intuitive and widely recognized. It has engineering mathematics that have been co-opted for management. Furthermore it is often the case that students have introductory exposure to Excel either in high school or in an introduction engineering course in “computer literacy.” In the College of Engineering at our University, all incoming College of Engineering freshmen are required to complete the introductory computing course, E115. This course’s purpose is to ensure the same level of computer competency among the students. Excel is a small portion of that course. Furthermore, almost every graduate will have access to Excel on their desktop computers when they go to work.

However there is a lot more to Excel than meets the eye of many engineers. It has numerous ways to express equations and the relationships among them, often without using the perplexing addressing methods. There are numerous functions for assisting in financial, date and time, math and trig, statistical, database, text, logical, information, and general engineering calculations. Its graphing facilities are impressive as is its data handling facilities. What is often overlooked is the modeling tools including optimization, sensitivity analysis, and scenario management. Features like data lists and Pivot table/charts provide enormous flexibility for summarizing and formatting data. Data can be imported and incorporated in the worksheets anywhere from legacy files to modern data bases. By adding input controls and data validation, worksheets can be designed to accept valid data for special uses from the design of amusement park fountains to sales engineering expense reports. Furthermore “add-ins” provide numerous facilities for statistical analysis, optimization, and database interfaces.

Finally, the “scripting language” of Visual Basic for Applications (VBA) provides “glue” that allows users to piece objects together from Excel and any other application on the Windows platform that exposes its object structure. Thus students can, through a simple programming language, become knowledgeable about objects and object properties including graphics. They can write custom functions and create libraries (for example to queuing formulas). They can build graphical user interfaces of forms and windows that accept data, import files and provide convenient navigation for the acquisition of data and the presentation of summaries, so important to decision support systems. Because VBA is generally incorporated in software applications that exist on the Windows platform, numerous software applications contain it. Once you learn VBA for Excel you can use VBA to extend Access, Word, PowerPoint, Visio, Outlook, and numerous other software packages like AutoCad and Arena. VBA is

actually a very simple language, but when used in the context of an object-based application like Excel becomes a powerful tool.

The use of Microsoft Excel with VBA is somewhat arbitrary and controversial. For example, a strong competitor to our choice of Microsoft Excel is MatLab, which offers many of the same features as Excel and other capabilities. However Excel is generally loaded on most personal computers and is a required option for computers used at our university. Furthermore it meets our needs for a modeling tool. It was felt that more companies have/provide Excel as part of their worker's computers. Therefore, Excel was chosen as our computing tool. We believe that the introductory course should be about one-third devoted to Excel and two-thirds devoted to the use and application of VBA.

Developing and Teaching a Computer-Based Modeling Course

While the design of the course is fundamental to its creation, the teaching and delivery of the course will determine the ultimate success. Bear in mind, this is one of the first engineering courses that a student takes during their college career. Therefore it is important to engage the students in learning about their discipline. However this engagement must be done in a way that permits multiple instructors and multiple sections to be taught to offer uniformity in computing experiences. "Scale" is important at lower level classes simply due to the number of students and the limited teaching resources. During the second offering (Fall 2006), a formal anonymous survey by our director of assessment was performed to ascertain the impact of our approach. The questionnaire represents a 90% return from both sections. The questions were designed to ask their confidence with Excel and VBA, opinions of the class, course components as well as homework and project assignments. A few informal surveys were given by the authors to get feedback during the semester.

Our new course is 3 hours of credit on a semester basis. Because of the in-class lab requirements (see next section), there needs to be recognition of the "laboratory" nature of the course. Traditionally, the Java and C++ course would be 2 hours of "lecture" and 3 hours of a separate "lab" to review programs. However, our course integrates the lecture and lab together since we have computers in the classroom. Consequently there is a combined four hours of lecture/lab. See the next few observations on how to conduct a two hour time period.

One of the drawbacks of the old programming course was the students learned about a computing without a computer to immediately try the concepts being taught. One of the authors uses the term "Immediacy Effect" when using student computers to teach. The students obtain immediate response to whether or not they correctly grasp the current concepts rather than waiting for later, such as at back in their rooms or in a separate lab later. In many cases, students get stuck on little things and you are there to help during class and not at home. To eliminate this problem, we have integrated the lecture and lab into a single time period where each student has a computer in front of them.

Many colleagues state that they will not teach with computers because the students have way too many distractions and they will not pay attention to what is being said by the instructor. Certainly technology can be misused by delivering lectures entirely by computer slides, by web pages or streaming video. These may make students even more passive than they are in normal classes⁴. If all you do is ask the students to look at your PowerPoint slides or you assume that they will take notes using the computer, you are definitely creating an environment in which students ignore you by doing homework, chatting with their friends via IM, emailing, playing video games or surfing the web. One can argue this could happen without computers via the use of cell phones.

Table 1 shows the data from the survey that deal with using computers in the classroom. It is clear that students strongly prefer having the computer in front of them with the same results across the two sections and two instructors. For question 25, over 90% disagree or strongly disagree allows the

students to be easily distracted owing to our teaching approach (See the next few Observations). The only anomaly was the responses from the Industrial Engineering (IE) section on questions 25 which was not as strong for having the computer in front of them but this contradicted in the other two questions which were asked for consistency.

Table 1: Questions Related to Having Computers in the Classroom

Rate your opinions about the class	IE Section					TE Section			
	Strongly disagree	disagree	Agree	Strongly agree		Strongly disagree	disagree	Agree	Strongly agree
25. I would prefer to use a laptop or computer in class rather than listen to the instructor.	3.85	50.00	26.92	19.23		13.04	17.39	34.78	34.78
26. Using a computer in class helps to keep me focused on the work.	0.00	7.14	64.29	28.57		0.00	20.83	45.83	33.33
27. I would learn as much if I took this class without a computer in front of me.	70.37	22.22	3.70	3.70		58.33	33.33	8.33	0.00
29. I prefer lecture and laboratory to be integrated together, so that I can practice what we learned in lectures at the same time.	10.71	17.86	53.57	17.86		4.17	0.00	54.17	41.67
32. Instructors should demonstrate what they are talking about by using a computer in class.	0.00	7.14	57.14	35.71		0.00	13.04	52.17	34.78
34. I would prefer to listen to the instructor rather than have hands-on computer activities in class.	23.08	53.85	11.54	11.54		16.67	54.17	12.50	16.67
35. If I have a laptop or computer in class I easily get distracted (e.g. send email messages, IM).	33.33	62.96	3.70	0.00		34.78	56.52	4.35	4.35

Active and Cooperative Learning

Rich Felder [2,3, 4] states that “Even if you're a real spellbinder, after approximately 10 minutes of straight lecturing you begin to lose a fraction of your students---they get drowsy or bored or restless, and start reading or talking or daydreaming. The longer you lecture the more of them you lose. Forcing them to be active, even if it's only for 30 seconds, breaks the pattern and gets them back with you for another 10-20 minutes.” It is well-known that when student participate in their learning, their learning is greatly enhanced. So it is important to engage the students so they are active and occupied during class. Engagement is especially important since they have this huge distraction of a computer in front of them.

One way to engage the student is using in-class assignments and exercises. We refer to these as “in-class labs” to convey the laboratory nature of these exercises. Essentially, the course revolves around the labs and there is very little formal lecture time. There is the occasional introduction to a modeling topic that will take 5-10 or so minutes of time, but even those are punctuated with examples that the students should or could implement and run. Lecture is more spontaneous since they arise from “teaching moments” which are instances during class when students realize they have a problem and now some commentary from the instructor is needed. At those times, students are most open to listening since they have an immediate use for the information. This approach might be called “just-in-time lecturing.” See <http://www.wintersim.org/upload/files/InClassLab02.pdf> which gives an example of one of the Excel In-Class Labs.

For the in-class lab to be effective, it needs to: (1) occupy the student during the entire two hour period, (2) challenge the student to critically think about their response, (3) produce questions about the modeling tool or approach, (4) seek help if they have a problem from the teacher, from a teaching assistant, or from another student, and (4) allow some flexibility for the instructor to use “teaching moments” to elaborate on specific issues.

Each lab consists of two to five different problems or case studies that need to be modeled. As can be seen from the example lab1, four different problems are used to illustrate different points for that day. It is known from instructional research that giving group exercises without accountability is not as effective^{3,6}. Therefore, the students are asked to answer a series of questions and turn them in to be

graded. This counts for 8-10 percent of their grade to emphasize the importance of them and to recognize the attendance of students (our university requires attendance to be taken for first-year classes). The following lists the contents of each lab.

- Beginning portion of the lab gives an overview of the problems and topics of the day
- Students are often instructed to download today’s spreadsheet from the website which may have data, code, etc. already available
- Each part of the in-class lab starts with background of the problem followed by a series of steps that have to be performed with more explanations when needed
- Intermixed with each of the steps is a series of questions that the students have to answer
- A subset of those questions are repeated on the front page, which have to be filled out and turned in at the end of the period

From question 35 of the survey seen in Table 1, the students do not have time to get distracted because they are completely engaged in the process of asking and answering questions. The class moves at very good pace since we keep them extremely engaged during the entire period. Table 2 shows more results directly related to the in-class labs and shows the students strongly prefer them. Even though the labs are completely self-contained and can be done without instruction, the students prefer to have lecture intermixed as along with getting guidance from the instructor (question 43). The one drawback of the labs is that over half the students feel there are too many questions. Therefore revisions to the labs need to address this issue without losing accountability that keeps them in engaged.

Table 2: Effectiveness of the In-Class Labs

Rate your opinions about the class	IE Section				TE Section			
	Strongly disagree	disagree	Agree	Strongly agree	Strongly disagree	disagree	Agree	Strongly agree
29. I prefer lecture and laboratory to be integrated together, so that I can practice what we learned in lectures at the same time.	0.00	7.14	42.86	50.00	4.17	0.00	29.17	66.67
34. I would prefer to listen to the instructor rather than have hands-on computer activities in class.	33.33	62.96	3.70	0.00	34.78	56.52	4.35	4.35
38. I prefer separate lectures with separate laboratory sessions to practice what we learned in lectures at my own pace.	25.93	40.74	18.52	14.81	29.17	50.00	12.50	8.33
43. I prefer to work through the labs on my own, then have the instructor go through them.	15.38	57.69	19.23	7.69	25.00	62.50	8.33	4.17
44. The labs consist of too many questions.	0.00	14.29	57.14	28.57	0.00	31.82	50.00	18.18
45. I like the step by step process of the labs	0.00	3.57	67.86	28.57	4.17	4.17	66.67	25.00

Limited Computer Resources

Computers are needed for in-class model construction. Learning to use the computer and its software is a clear imperative for the course. If each student has their own personal computer, then computing resources do not limit the limit size the class. To employ student-owned computers, there must be a university policy requiring their purchase. However today the expense of a well-equipped laptop is not the roadblock it was several years ago. Furthermore if the specifications are strict enough, the computing power of modern laptops is quite sufficient for a computer-based modeling course using Excel. Only when there are lengthy computations, such as needed for a series of simulation or a complex optimization, will computational time become an issue. We started the first offerings in our various computer teaching laboratories and are now moving towards “student-owned computers” which is an expectation by the College of Engineering at our university.

Limited Teaching Resources

The teaching resource is a more complex issue and one that is of prime concern to administrator who staff a difficult class. First, we need to state a bias. An experienced faculty member should be in

charge of the class and be the primary teacher. There is a tendency for introductory classes to be assigned graduate students and we think that is a poor policy. Computer modeling is not just learning to use a tool, like Excel. There is a modeling or engineering component that comes from experience and maturity with a discipline. Part of the responsibilities of this introductory course is to introduce the student to the systems engineering discipline. An experience faculty member can best describe how the models and the modeling apply to the discipline since they can bring their own experiences as examples into the course. Also, experience is needed in VBA programming. While VBA is a simple programming language, the teaching must convey the importance of programming style and program organization that only comes from years of experience.

A single faculty member could teach perhaps 50 or more students in a single room, if the students had their own laptops and the in-class lab style was used. However, there is a need for immediate help for the students. While it is possible for a faculty member to conduct a large classroom of students engaged in active learning using their own computers, there is the need for additional in-class help. Students can ask questions of the faculty member and of other students, but often these questions are specific to that student. The student perhaps can't copy a file, can't execute an operation, has encountered an error of unknown origin, etc. If the faculty member stopped for each such question, then there would be numerous interruptions and there would be minimum class progress. To aide students during the class and to maintain the class progress, we recommend that teaching assistants be employed.

Utilizing Technology to Assist Teaching

Using IM as Virtual Office Hours

Students today live in a technology world and instant message (IM) is ubiquitous.. How often have we as faculty sit in our office during the allotted time for office hours never seeing a single student? Often students have classes, have to work, have meetings, etc for reasons not coming to your office. Sometimes they may feel intimidated about asking questions face to face. For this type of course, the students are working on the computer when they get stuck, have a question, or need a clarification. Often it is a quick yes or no answer of that is what is meant in a homework statement and the student is able to continue again. Email has a lag that an answer to the question may lead to another question and so forth. Thus it may take several hours to answer, while IM may resolve it in a matter of minutes. The Industrial Engineering (IE) faculty member only uses traditional Email and office hours to answer questions electronically while the Textile Engineering (TE) TE faculty member uses both Email and IM to answer questions.

Question 48 of the survey asked the students "if using instant messaging to communicate with the instructor would be helpful in answering questions". The results indicate that 67% of the IE section believe instant messaging would be helpful in answer questions while 83% of the TE section state it is beneficial. Some qualitative responses from students of the TE section state they really like it.

- "I like that I can reach you with IM if your available for a quick question"
- "I like that you are on AIM to help us, sometimes you take a long time to respond. You should use an Away Message"
- "I like that you are available at night some times to answer questions when we get stuck"
- "I like to be able to IM so we can have a discussion since Email has a huge lag time."

Often faculty state they do not want to be in constant contact with students. A few things that we have learned over the past few semesters using IM to communicate:

- If you are busy then do not answer the IM or tell them you cannot answer for some specific amount of time

- Set guidelines of when you are available and not available (i.e., setup of virtual office hour times)
- Force the students to state their real name to identify themselves besides CrashBigDog684.

Utilizing a Tablet PC

The teaching medium has changed, as well as stayed the same, over the past few years. Starting with chalk boards we have migrated to overhead machines/slides, VCRS/DVDs, and to computers. The chalk board is still a significant delivery mechanism especially in Engineering where courses are laden with equations, graphs, and derivations are often more suited for this medium. The following reasons are often cited for not wanting to use a computer to deliver a lecture in Engineering

1. It is difficult to write equations on the computer in PowerPoint.
2. Can't modify an equation even if I have it in PowerPoint or some other electronic medium.
3. Can't respond to questions (i.e., need a graph, a new diagram, or need an equation based on students question) so therefore that I cannot perform inquiry guided learning.
4. I do not have time to put my notes into PowerPoint or Word.

Tablet PCs offer the advantage of using a computer but having a virtual chalk board since we can write on the screen using a pen [5, 6]. The Tablet PC eliminates many of the reasons cited. The Textile Engineering (TE) TE faculty member has been using a Tablet PC to teach for several semesters while the IE Industrial Engineering (IE) member had been teaching with computers but augmenting that with the chalkboard when questions or explanations arose. Often, people who use the blackboard and computer often have to redraw diagram or rewrite the equation before proceeding. The IE faculty member observed the TE faculty member using the tablet PC during the first offering and liked the efficiency and advantage of the approach. During the second offering the IE faculty borrowed a tablet for a portion of the class (until it died). This semester both faculty members are using Tablet PCs.

The faculty use the Tablet PC in a hybrid mode (i.e., they lay the screen down so they can write on it with the pen to make act like a chalk board but can use the keyboard and mouse like an ordinary computer). The advantage using a tablet PC to deliver a lecture is that you have the ability to directly write on the material presented, whether it is on a website or in a MS Word, Excel, or PowerPoint document. You can augment a claim for clarification or answer questions about the particular issues without having to rewrite the equation or redraw the graph. Figure 1 demonstrates answering a question about referencing by writing directly on the MS Excel spreadsheet to show the cell C4 row and column. The faculty members use this method to highlight elements of the spreadsheet, write the equations directly on the spreadsheet, or draw references between cells.

Figure 1: Using Tablet Penning in Excel

Figure 2 shows comments that were wrote on the actual MS Word document of the in-class Lab.



Figure 2: Using Tablet Penning in Word

Microsoft OneNote can be used to act as notebook paper to illustrate questions from the students or for clarification. Figure 3 shows an example of using MS OneNote as a virtual chalkboard to add some clarifications to questions that came up about the calculation of slope for two points.

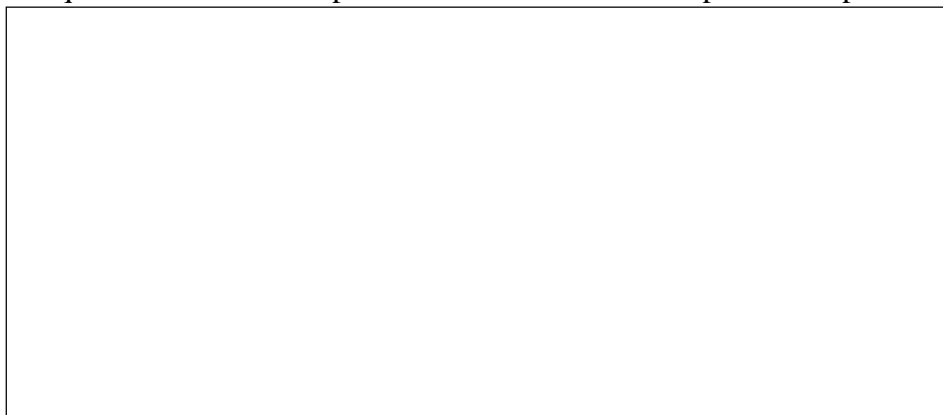


Figure 3: Free handing in Microsoft OneNote

As you can see, the faculty member drew a graph and wrote some equations based on the student responses. Figure 4 shows another example where there was a discussion of the Great Circle Distance and the instructor clipped part of a website that had a picture of the world. The great circle distance could be drawn onto this picture. You can imagine clipping parts of equations, pictures, programming code and commenting on it.



Figure 4: Clipping Website and Modifying IT

The newest versions of MS OneNote allow one to share their workspace which could allow students in the classroom to interact with the workspace (i.e., they could be asked to illustrate or label a diagram). One should start to use a Tablet PC incrementally. First, write in MS OneNote what you would write normally on the chalk board. Eventually, you will learn to add commentary to Excel, Word and PowerPoint documents. Question 41 of the survey was directly tied to the use of a Tablet PC. It is overwhelming that 91% of the Textile Engineering (TE) section agrees that using the Tablet PC is better while 57% of the Industrial Engineering (IE) section disagree with this statement. The disparity maybe due to the fact that the IE faculty member just started using the Tablet PC and only used it for a portion of the semester while the TE faculty member has been using the device for the past two years to teach.

Rate your opinions about the class	IE Section				TE Section			
	Strongly disagree	disagree	Agree	Strongly agree	Strongly disagree	disagree	Agree	Strongly agree
41. If your instructor used a Tablet PC was this more effective in teaching than using a computer and blackboard.	0.00	57.14	14.29	28.57	0.00	8.70	30.43	60.87

The following are few of the qualitative responses to a similar question.

- “The ability to access your notes after class is nice”
- “Easy to ‘see’ what you are referring to. Great teaching tool”
- “A chalkboard could be used but would not be as effective or efficient”
- “ You can magnify and change colors to highlight important things”
- “It is nice to be able to see both the program and the lesson at the same time with your comments on both.”
- “Yes great tool except legibility is sometimes hard to read”
- “...ability to write directly on the Excel document is very helpful. I think it saves time since you do not have to go to the board and redo everything”

Summary and Conclusion

This paper has presented an alternative to the first programming course that is generally offered as a service course and suffers from several limitations. The emphasis on the new course is developing the modeling capabilities of engineers and to teach the programming in the context of a problem. We have demonstrated the ability to utilize student owned computing effectively with our active learning In-Class labs and very little traditional lecturing. We feel this course will have a large impact on both our curricula and students and feel in the short run has been very successful.

Bibliography

1. Paper Authors, Example In-Class Lab <<http://www.wintersim.org/upload/files/InClassLab02.pdf>>
2. Felder, Richard, “Things I Wish They Had Told Me.” Chem. Engr. Education, 28(2), 108-109 (Spring 1994).
3. Collaborative Learning: Small Group Learning Page, <<http://www.wcer.wisc.edu/archive/c11/CL/default.asp>>
4. Cooperative Learning Papers by Rich Felder, <http://www.ncsu.edu/felder-public/Papers/Education_Papers.html#CL>
5. Using Tablet PC: A Guide for Educators, <<http://www.microsoft.com/education/edguidetabletpc.msp>>
6. Corrina Wu, Take a Tablet: Tablet Computers have the Potential to Redefine The Way Engineering is Taught, ASEE Prism, 16(4) (December 2006).