Appendix 3: COMPUTER BASED MODELING FOR ENGINEERS USING EXCEL AND VBA

Jeff Joines, Stephen Roberts & Dianne Raubenheimer

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 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 the two faculty members to develop this computer-intensive course. Of special interest is the classroom environment bolstered by the use of 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 s regarding the computing experience for engineers and suggest that there are other approaches to this course. We describe our own "solution."

A Bit of History

The computing course was introduced to the engineering curriculum sometime in the late 1950s and early 1960s. If you got your engineering degree before 1965, you probably took a Fortran programming class using something like an IBM 1620. You prepared your program on punched cards and "loaded" the cards along with the operating system components onto the machine.

Computing technology dramatically changed over time - from cards to terminals and magnetic tapes and from 16 bit machines to 64 bit machines (mainframes). Even the programming language evolved from Fortran II to Fortran IV to Pascal and Algol. Eventually that introductory computing course taken by all engineers evolved to today where it is workstations and personal computers on networks employing C++ and Java.

> ***Observation 1:*** *An initial programming course has been a part of early engineering education since computers become accessible to larger numbers of students.*

Students who took that initial programming course often found many ways to use the computer, especially in graduate school. Most faculty knew the one language (Fortran) that the students used and could give assignments. Therefore, the students continued to use their programming in subsequent courses providing continuity. Also since that was the only course often available, some engineering students in the earlier times could devote themselves to "computing" and it became a career for them.

Impact of Computer Science

In the late 1960s an intense interest in computing began to evolve and more faculty were showing considerable skill and interest in computing, to the point where curriculums were being developed that were identified as "computer science" and/or "computer engineering." Courses began to proliferate and, as the say "the rest is history." Today most colleges and university offer some kind of degree in computer science. It is a full-fledged discipline, some within engineering and some outside engineering. There are journals, meetings, research conferences, professional organizations, etc. Many graduates now identify themselves as computer scientists and computer engineers.

> ***Observation 2:*** *One course in programming no longer qualifies as expertise in programming.*

An engineer who desires expertise in programming will need to "minor" in computer science and will need take three or four computer science courses. It is no longer the case that one course and interest in programming qualifies you to become a computer scientist. It is now extremely rare that engineers take those kinds of jobs. Most of this is due to the extensibility of today's languages (Java, C++). The one "be-it- all" course that develops a student's algorithmic and problem solving skills along with learning a language is gone. The introductory programming course is offered as a service course by the Computer Science department at our University which is like most Universities. It is taught to a wide variety engineering disciplines at the same time. *It is taught the same way as it has been for many years.*

Engineers are Modelers

Observation 3: *Engineers are not programmers.*

Does this mean that engineers are not interested in using programming and other computing tools? Computing is more a part of engineering education than ever before. In fact, it is fundamental to an engineer's education. However, engineers utilize computing in a variety of ways. Probably most use software "applications" like ProE, AutoCAD, Maple, SAS, etc. If you need a program written, students may use C, C++, Java, and perhaps FORTRAN to tackle the problem. No doubt you may have a few students requesting to do the assignment in Excel, MatLab, or Mathmatica. Computing in engineering now covers an immense set of options. However there is a clear theme.

Observation 4: *Engineers are modelers.*

Engineering remains fundamentally a problem-solving endeavor. Computing is now a natural part of the problem-solving tool set. But is the introductory programming course meeting these needs (i.e., can Engineers use computing to solve problems)? For our two departments, the current computer science course was not meeting those needs.

While an introductory programming course may be useful, it is simply not enough for takers to learn all they need about programming. At best they learn to do procedural programming and the whole world of object-oriented concepts; graphical user interfaces, data structures, etc. are save for future courses. Yet applications within the engineering disciplines employ all these ideas. Part of the problem is the diversity of applications within engineering disciplines. Different disciplines have different computing needs. Some may need tools like Maple and MatLab, yet for other Excel and Access is prominently used. There are consequent implications for any introductory programming courses. Introductory courses are driven to fundamentals only.

Observation 5: *"One size (introductory programming course) does not fit all"*

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 courses 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.

Observation 6: *Systems engineers model decision processes.*

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
7. Industry constituents want our systems engineers to be able to tackle problems using appropriate computing.

The next section will discuss the development of a more appropriate computer-based modeling course while the following section will discuss the issues of *Teaching a Computer Based Modeling Course* with computers and tablet PCs. Finally, we will summarize and conclude our observations.

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.

> ***Observation 7:*** *Modeling and programming must be integrated into an introductory computing course.*

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 are often overlooked are 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.

> ***Observation 8:*** *Microsoft Excel is an ideal vehicle to teach modeling and programming (VBA) to understand the computer-based modeling of decision processes.*

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.

Textbook for Computer-Based Modeling with Excel and VBA

The textbook for a course often determines its content. Fortunately there are numerous books on Excel and VBA which is the good news. The bad news is that they are designed for the general user and not for systems engineering students. However for VBA, we have identified the following book that is oriented toward modeling and decision support systems:

> *VBA for Modelers, 2nd Edition, by S. Christian Albrigh.t*

For Excel we reference, but don't require the following book:

Excel 2003 Bible, by John Walkenbach, Wiley, 2003.

Consequently we have created our own set of notes for Excel that is specialized for our students. For example, topics include great circle distance using in logistics, ergonomic lifting, optimal order quantity, future value of compounding, data-based management, probability of revenues and costs, and simulation. These examples provide discipline-specific context for learning about worksheet navigation, entering complex formulas, use of named ranges, relative and absolute addresses, incorporating Excel functions, data validation, using solving, graphing, data tables, data import, lists, pivot tables and charts, debugging Excel formulas as well as developing decision supports systems with just Excel.

> ***Observation 9:*** *The course will not only introduce the students to computer-based modeling but will also be an introduction to the content of the discipline (systems engineering).*

Teaching and learning materials in Excel and VBA will need to be developed that highlight the content of the discipline. . The goal was to motivate problems students have seen in Physics, Chemistry, and Statistics as well as ones they will see in follow-on engineering courses. Therefore, we are introducing problem solving on problems that they will see and need to understand. We are treating many of the problems as equations/ black boxes where in later classes derivations and more understanding of the problems will be obtained.

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 (See Appendix A for the entire set of questions and results). 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.

Credit Hours, Computers, and Classroom Time

> ***Observation 10:*** *Two, two-hour class periods is sufficient to provide a useful computer-based modeling course.*

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.

Observation 11: *Teaching a computer-based modeling course with student computers can be quite effective and does not provide a distraction*

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 (See Observation 10).

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 classes[4]. 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 inwhich 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. Our solution will be addressed by the next observation (12).

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 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

	IE Section					TE Section			
Rate your opinions about the class	Strongly disagree	disgree	Agree	Strongly agree		Strongly disagree	disgree	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

General Responses from Student Surveys

- "Having computers has really helped me learn"
- "Not sure how you could take this class with out a computer in front of you"

Active and Cooperative Learning

Rich Felder[2] 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[3]. 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.

Observation 12: *In-class labs engage the student to be active in learning.*

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 lab[1], four different problems are used to illustrate different points for that day. It is known from instructional research that giving group exercises with out 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 require 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 Appendix A or 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 two 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

	IE Section					TE Section			
Rate your opinions about the class	Strongly disagree	disgree	Agree	Strongly agree		Strongly disagree	disgree	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

We have developed 27 in-class labs one for each of the time periods except for the tests.

> ***Observation 13:*** *Estimation of student's ability and time needed to perform programming tasks is crucial or give them snippets of code to get started.*

Active learning research states that the breakout exercises should be short and concise to be most effective [2, 3]. During the first offering of the course in Spring 2006, the two instructors would often ask the students to solve a problem which required independent thinking by applying the programming or modeling technique just discussed (e.g., a `For` loop). The student would have to perform the entire task by setting up all the variables, getting the information, setting the variable values, etc. as well as applying this new knowledge (of `For` loops) to finish the task. What we thought we were doing was to build on the previous exercises. What we thought would be beneficial in reinforcing topics and would be trivia,l turned out not be as effective. What was observed was that the students would get bogged down in the details of declarations of variables, message boxes or just typing all of this code into the editor, that they would miss the point of the learning exercise. We basically underestimated the time needed to perform the task. In the second offering, many of these sticking points were redesigned by giving the students more snippets of code that deal with "housekeeping". We could then focus on the concept we were trying to illustrate and the students would then just have to add the important pieces of codes. This was a enormous success in learning the material. However, looping still seems to be one of the poorest competencies in VBA as seen in Questions 14 through 23 which asks the students to rate their confidence in VBA in the results of the questionnaire in Appendix A.

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.

> ***Observation 14:*** *Using student-owned laptops provides sufficient computing capacity.*

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.

***Observation 15**: An experienced faculty member should be the primary teaching resource.*

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.

***Observation 16:** There should be one in class teaching assistant per 20-25 students.*

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.

The teaching assistants could be undergraduate or graduate students. Obviously undergraduate students tend to be "cheaper" on a per hour basis. Furthermore, by rewarding outstanding undergraduates with the opportunity to help "teach" a class they just had provides a strong morale boost. These undergrads easily communicate with their peers and often know student's problems from personal experience. They provide excellent out-of-class support as well. During the first offering with 28 students, one instructor didn't have in-class help and found it difficult to get around to all students needing help. Some students suffered from lack of attention, or the class got bogged and came to stand still. In later classes all sections employed in-class teaching assistants. The following table (Table 3) shows that having the TA in class is quite beneficial. It should be pointed out that the TE section has only ~70% benefit versus 90% for the IE section, which is a statistically significant difference ($F(1, 47)$, $p<0.0001$). The TA employed a new student who had just taken the course and is getting use to helping. We expect this to improve in future semesters.

Table 3: Effectiveness of Teaching Assistants

	IE Section					TE Section			
Rate your opinions about the class	Strongly disagree	disgree	Agree	Strongly agree		Strongly disagree	disgree	Agree	Strongly agree
42. Having the TA in class help with my	0.00	0.00	42.86	57.14		9.52	19.05	61.90	9.52

Thus the costs of the class have both a fixed and variable cost component. There is the fixed cost of the faculty member and the variable cost of the teaching assistants. Thus there are some economies to scale, but perhaps not what they are in other “big” lecture environments. The textile engineering faculty member has taught using computers for many years with smaller class sizes and been able to get assist students around the room. However larger classes impose other burdens.

Using IM as Virtual Office Hours

> ***Observation 17:*** *Using Instant Messaging (IM) to hold virtual office hours is very effective*

Students today leave 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 intimated 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 IE faculty member only uses traditional Email and office hours to answer questions electronically while the 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 the TE faculty member has learned over the past few semesters using IM to communicate:

- If you are busy then do not answer the IM or tell them you can not 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

Observation 18: *Using Tablet PCs as an effective delivery mechanism*

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 delivery 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 that I can 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. The Tablet PC eliminates many of the reasons cited. The TE faculty member has been using a Tablet PC to teach for several semesters while the 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.

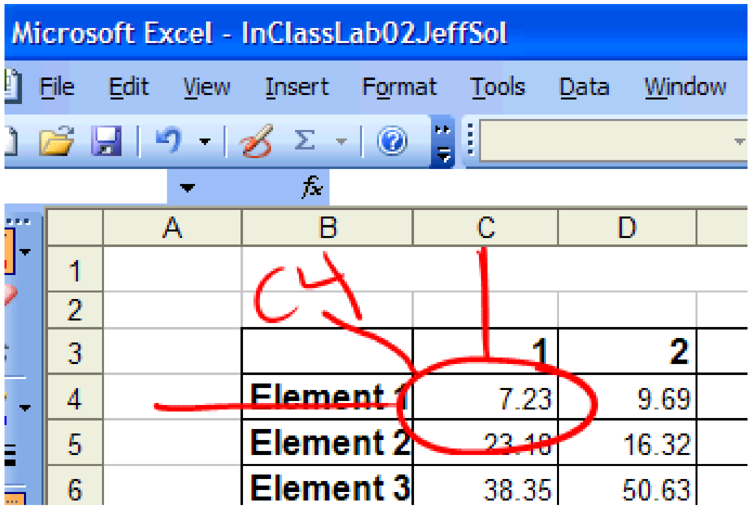

Figure 1: Using Tablet Penning in Excel

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

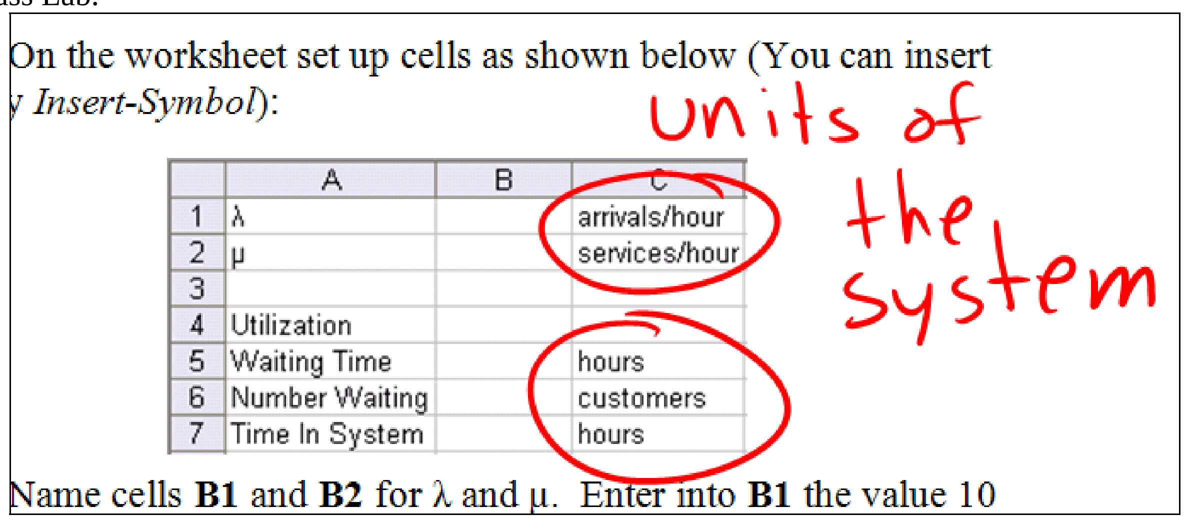

Figure 2: Using Tablet Penning in Word

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.

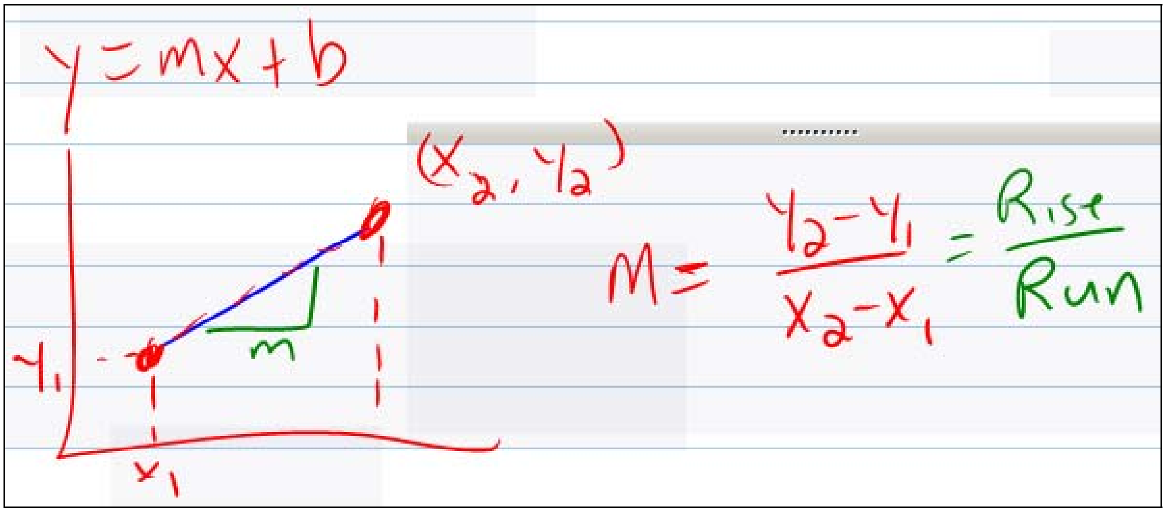

Figure 3: Free handing in Microsoft OneNote

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.

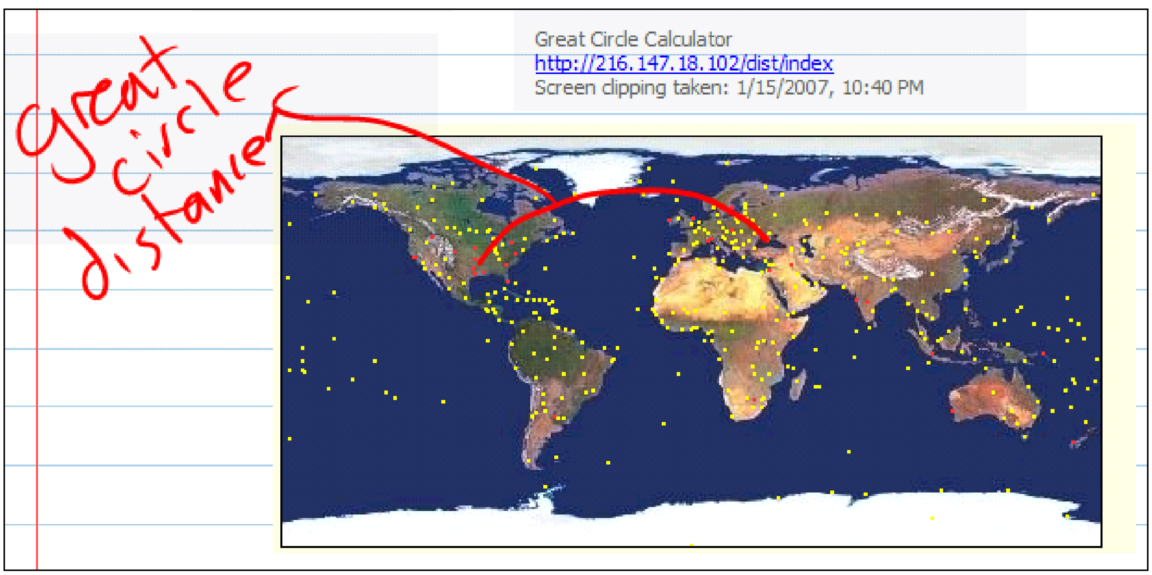

Figure 4: Clipping Website and Modifying IT

To get started using Tablet PC is to learn to use it 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 TE section agrees that using the Tablet PC is better while 57% of the 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. We will monitor this during the 2007 spring semester since all faculty will be using a Tablet PC.

	IE Section					TE Section			
Rate your opinions about the class	Strongly disagree	disgree	Agree	Strongly agree		Strongly disagree	disgree	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 that the TE section responded.

- “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”

Close Collaboration

Observation 19: *Close collaboration among faculty in teaching and developing of a course requires more time than teaching it alone*

Prepping a course for the first time requires a tremendous effort and research shows that you should not spend more than two hours for each credit hour[7]. The two architects of this new computer based modeling course have been friends for the past 16 years since one was a graduate student in the Industrial Engineering department. Both faculty have over 20 years of programming experience. More importantly both have been using programming to solve problems in both academia and industry. They have similar yet different experiences, styles and opinions of what is important. Since this class is the responsibility of two departments, the goal was to give the exact same experience whether a student took a section in the IE department or one in the TE department. The effort required complete collaboration and synchronization of the courses since we employed the same homework assignments, projects as well as gave common tests. The benefits of this collaborative process for student learning can be seen in the perceptions of students in both sections of the course (see tables 1 & 2). Even through there were two instructors with some differences in teaching styles, the student ratings were very similar in both sections. There were no significant differences in their ratings on the criteria presented in tables 1 & 2, nor were there any significant differences between sections in what they reported they had learned about using Excel and VBA (calculated using independent-samples t-tests).

It might be expected the course development in terms of lecture, homework, project and test creation would be easier for two people and possibly reduced by half. However, we have found that a true collaboration actually requires more time and energy. During the development of the 27 in-class labs, the following was observed.

- In development of the class the first time, each faculty may not have put in the same monumental effort if they were the only one involved.
- Fewer formal In-class labs would have been developed (i.e., thoughts may have been flushed out with informal topics going on during the class delivery).
- Ideas from one faculty member causes the other to include more ideas which caused more ideas from the other, etc.
- Often discussions of appropriate methods and solutions took place regarding key topics. Arguments for or against styles, topics, problems became a commonplace.
- The labs had to be complete and self contained since each faculty member would be teaching the material and they could not read the other one's thoughts. No free wheeling was to be allowed if the lab was not completely done.
- Learning occurred between each of the faculty members.
- We drove each other to reach perfection.

Collaborating would be by numerous emails day as well as by phone three to four times a day. They spoke directly after one would finish a lecture to find out what went well and wrong how the next person should try to fix it. Looking at the results from the survey (see Appendix A) administered to both sections, they are remarkable the same with just a few discrepancies indicating that a uniform experience had been achieved. Many of these differences can be explained. When looking at questions 22 and 23 which deal with building event handlers and developing decision support systems, ~60% of the TE section feel confident with the material as opposed to about ~35% in the IE section. There is a direct correlation between the TE faculty excitement for this material as well as experience.

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.

From question 50 of the survey, 90% of the students feel they have a learned a great deal in this class while over 55% plan to put on their resume that they are "Excel/VBA power user" with less than 20% stating they will not across both sections. Several of the students in the TE curricula are double majoring in other engineering curricula like biomedical, materials science or chemical engineering. Some of the feedback from these students is that they are excelling in their computer assignments in these other curricula while their peers struggle.

An interesting point is that over 85% of the students believe that homework was beneficial in their learning as seen in questions 31 and 40 of the survey. However, from comments made by the students they felt the homework was too hard and vague. We felt the homework should be demanding and require critical thinking. Often these young students were not use to thinking about how to solve a problem over a couple days. They were more accustomed to being given a series of steps and that if they complete them would be done. In many cases, the students are asked to analyze their solutions and respond to some general questions. This is a very intense and demanding course with homework due each Friday night at midnight. These homeworks are quite beneficial owing in part to the data and problems generally come from the faculty's research with industry. We feel that is preparing the students for their remaining engineering courses as well as when they go to work and they are asked to solve problems without clear guidance.

We are hoping that the third time will be less demanding on tweaking the In-class labs and only require the large effort in developing the homeworks and projects. The collaboration during the third semester has dropped as compared to previous two semesters. The Tablet PC has been shown to be an effective teaching medium in the context of this class.

Our goal is to administer a similar survey each semester as well as begin tracking these students for their rest of their college career and then into the beginnings of the workspace to determine if the computing as well as the critical thinking/ problem solving skills learned in this class had an impact. This impact will only be realized if these students are continually asked to use these skills in later classes in their curricula. An effort is underway to help faculty in later courses develop a few assignments/ projects that utilize Excel and VBA. In the TE curricula, two faculty have set in on the class to better understand Excel themselves so they can give these type of assignments.

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/cl1/CL/default.asp>

[4] Felder, R. *Technology-Based Instruction and Cooperative Learning*, ***The Interface*** (IEEE Education Society), August 2001, pp. 2–3.

[6] Cooperative Learning Papers by Rich Felder, <http://www.ncsu.edu/felder-public/Papers/Education_Papers.html#CL>

[7] Boice, *The New Faculty Member*, San Francisco, Jossey-Bass, 1992.

Appendix A: Quantitative Results of Survey

		IE Section					TE			
Textbook	**Mean**	**Seldom**	**Occassion**	**Often**		**Mean**	**Seldom**	**Occassion**	**Often**	
3. How often did you study the textbook?	**1.75**	42.86	39.29	17.86		**1.67**	50.00	33.33	16.67	
Rate your confidence with Excel	**Mean**	**Not Confident**	**Somewhat Confident**	**Confident**	**Very confident**	**Mean**	**Not Confident**	**Somewhat Confident**	**Confident**	**Very confident**
4. Moving arond the worksheet	3.71	0.00	3.57	21.43	75.00	3.79	0.00	0.00	20.83	79.17
5. Entering values and formula	3.71	0.00	3.57	21.43	75.00	3.79	0.00	4.17	12.50	83.33
6. Applying builf-in financial, statistical and math functions	3.36	0.00	21.43	21.43	57.14	3.29	0.00	16.67	37.50	45.83
7. Using goal seek	2.64	14.29	25.00	42.86	17.86	2.50	16.67	29.17	41.67	12.50
8. Uisng solver	3.04	3.57	21.43	42.86	32.14	3.00	0.00	37.50	25.00	37.50
9. Costructing data tables	3.32	0.00	14.29	39.29	46.43	3.17	0.00	16.67	50.00	33.33
10. Constructing graphs	3.57	0.00	7.14	28.57	64.29	3.65	0.00	4.35	26.09	69.57
11. Using pivot tables	3.25	3.57	14.29	35.71	46.43	3.38	0.00	8.33	45.83	45.83
12. Using lists	3.25	0.00	21.43	32.14	46.43	3.46	0.00	12.50	29.17	58.33
13. Using Named Ranges	3.68	0.00	3.57	25.00	71.43	3.83	0.00	0.00	16.67	83.33
Rate your confidence with VBA	**Mean**	**Not Confident**	**Somewhat Confident**	**Confident**	**Very confident**	**Mean**	**Not Confident**	**Somewhat Confident**	**Confident**	**Very confident**
14. Recording macros	3.86	0.00	3.57	7.14	89.29	3.83	0.00	0.00	16.67	83.33
15. Using Excel objects, methods and properties	3.21	3.57	7.14	53.57	35.71	3.33	0.00	8.33	50.00	41.67
16. Writing functions and subroutines	3.04	3.57	21.43	42.86	32.14	3.13	8.33	8.33	45.83	37.50
17. Defining variables of various types	3.32	0.00	14.29	39.29	46.43	3.25	8.33	8.33	33.33	50.00
18. Making assignments	2.93	0.00	32.14	42.86	25.00	3.13	8.33	16.67	29.17	45.83
19. Creating loops	2.57	10.71	42.86	25.00	21.43	2.83	20.83	16.67	20.83	41.67
20. Using 'ifs' and 'cases'	3.04	7.14	14.29	46.43	32.14	3.17	8.33	16.67	25.00	50.00
21. Creating your own forms and controls	3.11	3.57	10.71	57.14	28.57	3.50	4.17	8.33	20.83	66.67
22. Writing event handlers	2.29	17.86	46.43	25.00	10.71	2.71	16.67	25.00	29.17	29.17
23. Developing decision support systems	2.26	14.81	55.56	18.52	11.11	2.75	16.67	25.00	25.00	33.33

		IE Section						TE Section			
Rate your opinions about the class	Mean	Strongly disagree	disgree	Agree	Strongly agree		Mean	Strongly disagree	disgree	Agree	Strongly agree
24. I'll need a firm mastery of Excel/VBA programming for my future work.	2.83	4.35	30.43	43.48	21.74		3	10.53	10.53	47.37	31.58
25. I would prefer to use a laptop or computer in class rather than listen to the instructor.	2.62	3.85	50.00	26.92	19.23		2.91	13.04	17.39	34.78	34.78
26. Using a computer in class helps to keep me focused on the work.	3.21	0.00	7.14	64.29	28.57		3.13	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.	1.41	70.37	22.22	3.70	3.70		1.50	58.33	33.33	8.33	0.00
28. I am able to get help from my friends in class when I have questions or problems.	2.79	10.71	17.86	53.57	17.86		3.33	4.17	0.00	54.17	41.67
29. I prefer lecture and laboratory to be integrated together, so that I can practice what we learned in lectures at the same time.	3.43	0.00	7.14	42.86	50.00		3.58	4.17	0.00	29.17	66.67
30. Large lecture classes are best for my learning.	1.81	26.92	65.38	7.69	0.00		1.95	23.81	57.14	19.05	0.00
31. Homework assignments were helpful to my learning.	3.29	0.00	7.14	57.14	35.71		3.22	0.00	13.04	52.17	34.78
32. Instructors should demonstrate what they are talking about by using a computer in class.	3.50	0.00	7.14	35.71	57.14		3.67	0.00	0.00	33.33	66.67
33. Figuring out Excel/VBA programming problems does not appeal to me.	2.12	23.08	53.85	11.54	11.54		2.29	16.67	54.17	12.50	16.67
34. I would prefer to listen to the instructor rather than have hands-on computer activities	1.70	33.33	62.96	3.70	0.00		1.78	34.78	56.52	4.35	4.35
35. If I have a laptop or computer in class I easily get distracted (e.g. send email messages, IM).	2.07	18.52	59.26	18.52	3.70		2.29	12.50	50.00	33.33	4.17
36. It is a good idea for instructors to assign in-class individual work on computers.	2.93	0.00	22.22	62.96	14.81		2.63	12.50	25.00	50.00	12.50
37. Working with others in small groups in class using computers is not a good idea.	2.15	19.23	53.85	19.23	7.69		1.91	30.43	52.17	13.04	4.35

		IE Section						TE Section			
38. I prefer separate lectures with separate laboratory sessions to practice what we learned in lectures at my own pace.	2.22	25.93	40.74	18.52	14.81		2.00	29.17	50.00	12.50	8.33
39. Smaller classes with about 20 students in them are better for my learning.	3.23	0.00	7.69	61.54	30.77		3.38	0.00	8.33	45.83	45.83
40. The homework seemed unrelated to in-class activities and projects	2.18	14.29	64.29	10.71	10.71		1.71	41.67	45.83	12.50	0.00
41. If your instructor used a Tablet PC was this more effective in teaching than using a computer and blackboard.	2.71	0.00	57.14	14.29	28.57		3.52	0.00	8.70	30.43	60.87
42. Having the TA in class help with my learning.	3.57	0.00	0.00	42.86	57.14		2.71	9.52	19.05	61.90	9.52
43. I prefer to work through the labs on my own, then have the instructor go through them.	2.19	15.38	57.69	19.23	7.69		1.92	25.00	62.50	8.33	4.17
44. The labs consist of too many questions.	3.14	0.00	14.29	57.14	28.57		2.86	0.00	31.82	50.00	18.18
45. I like the step by step process of the labs	3.25	0.00	3.57	67.86	28.57		3.13	4.17	4.17	66.67	25.00
46. The homework assignments were about the correct level of difficulty.	2.32	25.00	25.00	42.86	7.14		2.67	16.67	29.17	25.00	29.17
47. The projects were useful in learning the material.	2.93	7.14	3.57	78.57	10.71		3.25	0.00	12.50	50.00	37.50
48. Using Instant Messaging (IM) to communicate with your instructor would be helpful in answering questions.	2.79	4.17	29.17	50.00	16.67		3.35	4.35	13.04	26.09	56.52
49. I feel that I received help when I needed it.	2.82	3.57	10.71	85.71	0.00		3.25	4.17	16.67	29.17	50.00
50. I feel I have learned a great deal in this class.	3.36	3.57	0.00	53.57	42.86		3.46	0.00	12.50	29.17	58.33

		IE Section					TE Section		
Course components:									
How do you feel about the following components of the course?	**Mean**	**I'd like fewer**	**The amount was right**	**I'd like more**		**Mean**	**I'd like fewer**	**The amount was right**	**I'd like more**
Homework assignments	1.32	67.86	32.14	0.00		1.46	54.17	45.83	0.00
Projects	1.93	10.71	85.71	3.57		2.00	4.17	91.67	4.17
In-class labs	1.61	39.29	60.71	0.00		1.88	16.67	79.17	4.17
Resume	**Mean**	**No**	**Don't know**	**Yes**		**Mean**	**No**	**Don't know**	**Yes**
Do you plan to state you are an Excel/VBA "power user" on your resume?	2.25	25.00	25.00	50.00		2.46	12.50	29.17	58.33
Would you like a job that requires you to use	2.43	14.29	28.57	57.14		2.13	33.33	20.83	45.83
Would you like a job that requires you to use	1.93	42.86	21.43	35.71		2.00	33.33	33.33	33.33