A “Boot Camp” for Preparing Freshman Students for Success in a First Course in Computing

Ken Christensen[1], Darrel Davis[1], and Dewey Rundus[1]

Abstract

A key course in the University of South Florida engineering core is *Computer Tools for Engineers* (EGN 2210). Enrollment ranges from 120 to 240 students per semester. The attrition in this course is typically one-third of the initially enrolled students. The experience level of entering students range from first exposure (to computers) to near expert. As an experiment in reducing attrition and improving the academic success of all students in EGN 2210, an intensive first-week “boot camp” was developed and tested. The boot camp allows students with little or no computing background to “come up to speed” early in the semester. Enrolling in boot camp is voluntary. The success of boot camp versus non boot camp students is being tracked. Initial findings from Summer and Fall 1999 show that boot camp students are statistically better than non boot camp students, but this appears to be due to self-selection. This self-selection of better students suggests the need for targeting of selected groups of students in future semesters if reducing attrition is a goal that is to be met.

1. Introduction

At the University of South Florida a course in computer tools, with an emphasis on FORTRAN programming, is taught to all engineering students. With the use of computers pervasive across all disciplines, the successful mastery of computer tools is important. The design and implementation of *Computer Tools for Engineers* (EGN 2210) is described in [1] (the course contents are on the web at [2]) and is only briefly overviewed here. The course is three credit hours, contains both lecture and lab, and is intended to be taken at the freshman or sophomore level (however, many students are juniors or seniors). The course is organized into five major topics, they are:

1. *Formula crunching* - use of a mathematics package (Mathcad)
2. *Number crunching* - use of a spreadsheet (Excel)
3. *What's under the hood* - computer internals (Von Neumann architecture)
4. *Think it through* - methods of design (flow charting, divide-and-conquer, and successive refinement)
5. *Just program it* - programming with a high-level language (FORTRAN)

FORTRAN programming is about 60% of the course and is covered using [11] as the textbook. Entering students have a broad range of backgrounds, perhaps more so than for any other engineering core course. Some students have programmed “all their lives”; others have never done much more than surf the web and simple word processing. Successfully reaching all of the students in EGN 2210 is a challenge. In the past 4 years, the attrition rate in EGN 2210 has been nearly one-third of the class. To address this attrition rate, and otherwise improve the level of learning for all students, a first week “boot camp” was proposed [13] and developed[2]. This paper describes the boot camp and an evaluation of its impact on student academic success over two semesters.

Recent changes in the engineering core at USF have changed the scope of EGN 2210 and now place the requirement for teaching computing on the individual departments (e.g., Electrical Engineering has implemented their own first course in computing based on MATLAB and C/C++ programming [8]). These changes to the engineering core will significantly change the content and focus of EGN 2210 beginning in Fall 2000.

---

[1] Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620.

[2] This work was funded by an internal grant from the USF Center for Teaching Enhancement.

The remainder of this paper is organized as follows. Section 2 reviews previous work in innovative methods for teaching computing. Section 3 describes the design and implementation of the boot camp for EGN 2210. Section 4 describes the evaluation of the boot camp. A summary and future work is in Section 5.

2. Existing Work in Innovative Methods for Teaching Computing

Much work has been done in studying effective ways to teach fundamental concepts in computer programming to beginning students [10]. Of interest to our work are new ideas in delivery and pedagogical methods.

Perkins et al. [12] develop a concurrent metacourse for supporting a high school first course in BASIC programming. The nine-lesson metacourse taught the students thinking and learning heuristics specialized for learning programming. A statistical study on an experimental group of 132 students with control group of 239 students show that the students who have taken the metacourse generate significantly fewer programming errors.

Macfarlane and Mynatt [9] present findings from a study done on the use of an Advanced Organizer for teaching Pascal. Advanced organizers are general concepts to bridge the gap between what is known and what is new. In the study, an experimental group of 118 students voluntarily signed up for the course and were given:

1. A tutorial manual which had statements necessary to perform tasks example array declarations
2. A training program which contained Turbo Pascal tasks
3. A two-part posttest designed to test the students' semantic and syntactic knowledge
4. Two forms of an organizer booklet

Students were placed into groups of 10 to 17 per session, each session lasting about 50 minutes. The results showed that the advanced organizer group did significantly better on tasks requiring semantic knowledge, but were on par when syntactic knowledge was tested.

In [6] Jackson addresses problems specifically associated with teaching concurrency. His method uses an introductory mini-course covering five 45-minute sessions and within that time frame five key concepts are covered (process concepts, concurrency in Ada, problems of concurrency, advanced programming, and task termination and deadlock). This introductory course lasted about two weeks and the general reaction from students was that the sessions were interesting and informative. No formal evaluation of impact to student learning is presented

Dorin [5] develops a workshop designed to improve students' programming skills and techniques. The workshop is offered between the programming and data structures courses in a computer science curriculum meeting for 75 minutes, twice a week, for 15 weeks. A representative set of the content includes component testing, abstract data types, discrete simulation, dynamics programming, backtracking, divide-and-conquer, numerical techniques, and dealing with very large integers. No formal evaluation of impact to student learning is presented.

3. The Design and Implementation of Boot Camp

Boot camp is designed to be conducted over the span of several evenings, or one weekend day, in the first week of class. Boot camp focuses on the programming (FORTRAN) portion of EGN 2210. The goal is to give students a very fast-paced jump-start on the course. The option to "enlist" in boot camp is voluntary. Part of the evaluation of boot camp, covered in Section 4, focuses on the possible self-selection process of this voluntary participation. The learning objectives for boot camp are:

1. Be able to enter and compile a short FORTRAN program
2. Be able to identify and fix simple syntax errors in an entered FORTRAN program
3. Understand the difference between variables and constants
4. Be able to identify and use INTEGER, REAL, and CHARACTER variable and constant types
5. Be able to write a FORTRAN program that uses built-in functions (e.g., SIN(), SQRT(), etc.)
6. Be able to write a FORTRAN program to input values, add them, and then output sum
7. Be able to identify if, loop, and subprogram structures in a FORTRAN program
8. Be able to write a FORTRAN program that uses IF statements
9. Be able to write a FORTRAN program that uses a DO loop statement
10. Be able to write a FORTRAN program that uses a user-written function

At the completion of boot camp, a certificate of completion is handed-out to the students during a lecture session containing all enrolled students.

3.1 Programming environment used in EGN 2210

The programming environment used in EGN 2210 is DOS-based and uses the WATCOM WATFOR-77 compiler, Version 3.0 (see [3]). The WATFOR-77 compiler is a small (single diskette) command-line compiler that fully supports FORTRAN 77 (ANSI X3.9-1978). The WATFOR-77 compiler also supports extensions including WHILE-UNTIL constructs. The language extensions are not used in EGN 2210. USF has a license agreement with WATCOM allowing for distribution of the compiler to students. The WATFOR-77 compiler is an old (mid-1980's) compiler and runs as well on an 80386 PC as on a modern Pentium-III PC. This is beneficial for students that do not have the latest technology at home. The WATFOR-77 compiler includes a built-in text editor, however one of the instructors (Christensen) prefers to use a stand-alone text editor (DOS edit) to teach students. By using a stand-alone editor, two DOS windows can be kept open on a Windows 9X/NT PC - one for editing and a second for compiling.

3.2 Implementation of Boot Camp

The Summer 1999 boot camp session was implemented over a 3-day period in the evenings. In Fall 1999, boot camp was implemented on a Saturday morning from 8am to 12pm. The summer boot camp covered all of the class topics, whereas the fall boot camp focused only on FORTRAN programming. For both summer and fall, boot camp was taught in a 40 seat lab with one PC per seat. The screen on the instructor's PC (at the front of the room) is displayed on a 20 monitors interspersed between the 40 PC's. This allows students to see exactly what the instructor is doing at all times.

In the summer boot camp, the time allotted for each topic was based on its relative importance in the actual course. Basic computer skills, Microsoft Excel and Mathcad were each given 1 hour, and FORTRAN was presented in the final 6 hours. Due to time constraints, the fall boot camp was a single intensive 4-hour session covering only FORTRAN. Students were coding their own programs within 10 minutes of the start. Figure 1 shows the first program (borrowing from the tutorial introduction in the classic "K&R" C book [7]). Getting this first program to run requires the use of an editor (to enter and save the program) and a compiler (to compile the program). The focus in this first program is not on the program itself, but in learning to work in the programming environment.

```
C *** My first FORTRAN program
      PRINT *, 'HELLO WORLD'
      END
```

Figure 1 – The first FORTRAN program in boot camp

The pedagogical style of boot camp is one of fast-paced concept/example/exercise sequences (each sequence is about 10 minutes). Extended lecture is avoided. The goal is for student to "do" and not just listen. Students are presented with a concept and an example of the application of that concept. They are then expected to extend the concept to solve a slightly more difficult exercise. When studying the concept of simple arithmetic (having already covered the concept of variables), Figure 2(a) shows the example program and Figure 2(b) the desired solution to the exercise, "Change the example program to increment your age by one and output, 'Soon I will be ___' where ___ is the incremented age." The exercise is written on the board, the solution (after about 10 minutes of working hands-on with the students) is finally completed on the instructor PC and thus displayed on the monitors for all students to study.

```
C *** Example of a variable
      INTGER AGE
      AGE = 20
      PRINT *, 'I am ', AGE
      END
```

Figure 2(a) – Example program

```
C *** Exercise solution for increment
      INTEGER AGE
      AGE = 20
      AGE = AGE + 1
      PRINT *, 'Soon I will be ', AGE
      END
```

Figure 2(b) – Solution for variable increment exercise

The sign-up for the boot camp is strictly voluntary and this introduced the problem of having students of varying competency levels in one setting. Students who are already familiar with some of the material often become bored, distracted, and even a distraction to others. To prevent this, boot camp is designed to be extremely interactive. Students have the freedom to question and react to material, and an overall positive attitude is built into the delivery of the content. Peer help is encouraged. This provides an atmosphere where guided and cooperative discovery learning could simultaneously take place, albeit at a very fast pace.

3.3 Development of training material for future boot camp instructors

One goal of the teaching grant [13] used to develop boot camp and fund one of the authors (Davis) to deliver boot camp in Summer and Fall 1999 was to also develop instructor training materials. This would enable boot camp to be offered in later semesters without, it is hoped, additional resources beyond the normally assigned teaching assistants. The content of boot camp was authored onto an interactive tutorial CD-ROM using the DemoShield [4] multimedia authoring software program. This training tutorial contains the actual content for boot camp along with general course ideas to aid the instructor. Although the primary focus is the instructor, the training tutorial can also be used by a student as a self-study guide. The CD-ROM contains:

1. General readme – how to use the tool
2. Course material – all the content covered in the boot camp
3. Support material – description of the package and how to use the hardcopies and evaluations
4. Instructor tips – useful tips to make the boot camp experience memorable
5. Credits – acknowledgments of people involved in the project

The CD-ROM is a part of a package that contains hardcopies of objectives, course guidelines, evaluations and an errata sheet. The plan is to have a record of every boot camp noting what works and what doesn't, paying special attention to previous data so that each boot camp is better than the previous. The package is designed to require minimum maintenance, but does allow for adjustments where necessary.

4. Evaluation of the Impact of Boot Camp

The goal of the evaluation process was to determine the effects, if any, of boot camp on academic success in EGN 2210. Enrollment in boot camp was voluntary. It is very possible that boot camp students are self-selected from a more motivated and harder working student group than average (i.e., that boot camp students are "better students"). This would mean that the students enrolling in boot camp would do better in EGN 2210 regardless of the content or delivery of material in the actual boot camp. This is no different than saying that it is the ability to get into an ultra-competitive school (e.g., Harvard MBA school) that may be a bigger determinant of future success than what is necessarily learned in the school. Eliminating the self-selection bias, if any, is a key factor in the design of the evaluation methods. However, self-selection may also have a very positive social value if it can be shown that boot camp students do better, then in future semesters students who might otherwise not (i.e., if not given a "proof") be motivated to participate will enroll. It is very likely these less motivated students that can gain the maximum benefit from boot camp, not only for EGN 2210, but also in developing better disciplined and proactive study habits for all of their courses.

4.1 Design of the evaluation methods

A survey, first employed in Fall 1999, was designed to collect information on the background of all students in EGN 2210 and boot camp. The Appendix contains the actual survey given both in class and in boot camp. Included in the Appendix are summarized survey data from Fall 1999. Using the survey results, groups of similar students in boot camp and not in boot camp can be isolated. The non boot camp students are the control groups used to measure as much as possible, the unbiased effects of boot camp. Five student groupings were made based on the following parameters:

1. Current GPA at USF
2. Community college transfer student status
3. Intended major

4. Previous programming experience
5. Grade on exam #1 (for Fall 1999 only)

The fifth grouping for Fall 1999 involved using the exam #1 score (there are two exams in the course). Exam #1 covers material explicitly not covered in boot camp (i.e., exam #1 has no FORTRAN content). Thus, any learning in boot camp should have no effect on exam #1 scores making this an independent parameter for group selection.

For Summer 1999 we failed to take a pre-class survey and are thus unable to determine student background when entering the course. The data gathered during the summer was the final course grade of boot camp participants and non-participants. No intermediate grade data (e.g., quiz grades) was collected. A hypothesis test was posed where the probability of success was defined as achieving a cut-off grade of B in the class. The hypothesis test was repeated for a cut-off grade of C. The hypothesis was the boot camp students are more successful than non boot camp students.

For Fall 1999, the survey results and tracking of intermediate grades is being done. To date, three significant data points have been tracked: quiz #1, quiz #2, and exam #1. Subsequent quizzes and exams will be on boot camp material. The grades for quiz #1 and #2, and exam #1 were analyzed, but only the exam #1 grades were used in a hypothesis test that boot camp students are less able the non boot camp students.

4.2 Evaluation results from Summer 1999

In Summer 1999, EGN 2210 was taught by three graduate students as four separate sections of about 40 students each. No survey was given at the beginning of the semester. Of a total of 150 students enrolled, 32 participated in boot camp. The final average course grade (out of 4.00) for boot camp participants was 3.06 and 2.99 for non-participants. Table 1 shows the number of boot camp and non-boot camp students with a final course grade of B and C where $\hat{p}_1$ is the success rate for boot camp participants and $\hat{p}_2$ for non boot camp participants.

Table 1 – Grade data for Summer 1999 EGN 2210

	Final grade of B	**Final grade of C**
Boot camp	26 of 32	28 of 32
Non boot camp	87 of 118	103 of 118
$\hat{p}_1$ **and** $\hat{p}_2$	0.8125 and 0.7373	0.975 and 0.873
ratio ($\hat{p}_1/\hat{p}_2$ **)**	1.102	1.002

A 95% confidence hypothesis test that boot camp students are more successful results in a range of –0.082 to 0.232 for a grade of B and 0.127 to 0.132 for a grade of C. This range is computed as,

$$(\hat{p}_1 - \hat{p}_2) \pm z_{\frac{\alpha}{2}} \cdot SE(\hat{p}_1 - \hat{p}_2) \tag{1}$$

where z is the standard Z-score for 95% confidence (1.96) and $SE()$ is the standard error function computed as,

$$SE(\hat{p}_1 - \hat{p}_2) = \sqrt{\frac{\hat{p}_1(1-\hat{p}_1)}{n_1} + \frac{\hat{p}_2(1-\hat{p}_2)}{n_2}} \tag{2}$$

where n_1 and n_2 are the number of boot camp and non boot camp participants, respectively. If the confidence interval range crosses zero, then the hypothesis cannot be supported. Thus, there is no statistical evidence of boot camp affecting student performance (as measured by final course grade) in EGN 2210 for Summer 1999. In any case, without knowing the initial level of the students participating (and, equally, not participating) in boot camp, it cannot be stated that boot camp had any effect (either positive or negative) on the performance of students.

4.3 Evaluation results from Fall 1999

In Fall 1999, EGN 2210 is being taught by one of the authors (Christensen). The Fall 1999 course has 120 students enrolled with 38 having participated in boot camp. At the point of exam #1, 12 students failed to take the exam #1. These student represent an 10% attrition level at less than the mid-point of the semester. Of the 12 already failed students, only one was in boot camp. Table 2 shows the average quiz and exam grades for both boot camp and non boot camp participants. Grades are computed in two ways, 1) omitting the "no-show" students (*omit*), and 2) using a 0 for the quiz or exam grade of a no-show student (*no omit*). The *ratio* row shows boot camp divided by non-boot camp.

Table 2 – Grades for non-boot camp material in Fall 1999 EGN 2210

	Quiz #1 (omit)	**Quiz #1** (no omit)	**Quiz #2** (omit)	**Quiz #2** (no omit)	**Exam #1** (omit)	**Exam #1** (no omit)
Boot camp	76.2 %	74.2 %	93.5 %	86.1 %	78.7 %	76.8 %
Non boot camp	66.9	57.9	83.3	65.0	74.8	64.8
Ratio	1.1	1.3	1.1	1.3	1.1	1.2

Using a cutoff grade of B and the same analysis method of Section 4.2 using equations (1) and (2), the confidence interval was 0.4266 to 0.0523 for omit and 0.3676 to –0.0235 for no omit. Using a cutoff of C, the interval was 0.3223 to 0.0007 for omit and 0.3934 to 0.0675 for no omit. Another test looked at the probability of students skipping quizzes and/or tests, and the data revealed that the probability of a non-participant missing a quiz or exam #1 was Pr[miss] = 0.4109, which was more than three times greater than a participant with Pr[miss] = 0.1267. In the case of grades based on academic year, boot camp students did on average 20% better. When sorting was done based on the responses to computer literacy, boot camp students did on average 15% better. <u>These statistics show, with high confidence, that the students enrolling in boot camp are academically better students than those that did not enroll.</u>

At the end of the semester, a final evaluation of boot camp will be done. The goal of this final evaluation is to conclusively determine if boot camp actually improves student performance, or is the better academic performance due only to self-selection. Control groups will be determined based on the five parameters described in Section 4.1. For example, we will match boot camp and non boot camp students with similar (as best possible) exam #1 grades and then do a paired test for the hypothesis that boot camp improves the final grade in EGN 2210.

5. Summary and Future Work

We have developed and implemented an introductory short course, or boot camp, with the goal of improving academic success of students in *Computer Tools for Engineers* (EGN 2210). One goal of the boot camp is to reduce the current attrition rate of one-third the initially enrolled students. Boot camp was advertised to the students in EGN 2210 as a means of quickly coming-up to speed. It was stressed that students with little, or no, experience should enroll. However, our evaluation results show that instead of the weaker students enrolling, better students enrolled. In other words, boot camp cannot reduce attrition without a much more forceful targeting of the less experienced (and academically weaker) students.

Future work will entail completing the evaluation of Fall 1999 boot camp impact on student success. The hypothesis to be tested is that boot camp improves the final grade over "similar" (in background) students that did not participate in boot camp. For future semesters we hope to take a survey of student background in the beginning of the first class, tally the results during the class (e.g., by a teaching assistant), and then students with survey "scores" lower than the class average will be – at that time - required to enroll in boot camp. The impact of this forced approach will then be evaluated against the voluntary boot camp program.

Acknowledgements

The authors would like to acknowledge the USF Center for Teaching Enhancement for funding this work. The funding allowed for the support of one of the authors (Darrel Davis, a USF Computer Science and Engineering graduate student) to implement and evaluate boot camp.

List of References

1. K. Christensen and D. Rundus (1998), “A First Course in Computing for Engineers,” *Proceedings of the 1998 ASEE Southeastern Section Meeting*, pp. 247 - 255.

2. K. Christensen (1999), “Computer Tools for Engineers - EGN 2210 - Fall 1999,” URL: *http://www.csee.usf.edu/~christen/class7/class7.html*.

3. G. Coschi and J. Schueler (1989), *WATFOR-77 User's Guide, 5th Edition*, WATCOM Publications Limited, Waterloo, Canada.

4. DemoShield software (1999), URL: *http://www.installshield.com/demoshield/*.

5. P. Dorin (1997), “Practice + Paradigms: Experience with a First-Year Programming Workshop,” *ACM SIGCSE Bulletin*, Vol. 29, No. 4, pp. 42 - 44.

6. D. Jackson (1991), “A Mini-Course on Concurrency,” *ACM SIGCSE Bulletin* Vol. 23 No. 1, pp. 92 – 96.

7. B. Kernighan and D. Ritchie (1988), *The C Programming Language, Second Edition*, Prentice-Hall Software Series, Englewood Cliffs, New Jersey.

8. F. King (1999), “Electrical Engineering Computer Methods - EEL 4937 - Fall 1999,” URL: *http://www.eng.usf.edu/~king/eecomp/eecomp.html*.

9. K. Macfarlane and B. Mynatt (1988), “A Study of an Advanced Organizer as a Techniques for Teaching Computer Programming Concepts,” *ACM SIGCSE Bulletin*, Vol. 20 No. 1, pp.240 - 243.

10. R. Mayer (editor) (1988), *Teaching and Learning Computer Programming, Multiple Research Perspectives*, Lawrence Erlbaum Associates, Publisher, Hillsdale, New Jersey.

11. L. Nyhoff and S. Leestma (1996), *FORTRAN77 for Engineers and Scientists*, Prentice Hall, Upper Saddle River, New Jersey.

12. D. Perkins, S. Schwartz, and R. Simmons (1988), “Instructional Strategies for the Problems of Novice Programmers,” Chapter 7 (pp. 153 – 178) of R. Mayer (editor), *Teaching and Learning Computer Programming, Multiple Research Perspectives*, Lawrence Erlbaum Associates, Publisher, Hillsdale, New Jersey.

13. D. Rundus and K. Christensen, (1998) “A Technology-Supported “Boot Camp” for *Computer Tools for Engineers* (EGN 2210),” internal grant from the *USF Instructional Development Grants Program*. Grant proposal is available by request from the author.

Appendix - Survey to Measure Student Background

This is the student survey that used to collect information on student backgrounds. The identical survey was given to the entire class and to the boot camp class. The summarized survey results below are for Fall 1999 and are for only those students who took the survey. In addition, students did not always answer all questions. Key summary statistics for Fall 1999 EGN 2210 are:

- Initial enrollment of 120
- 38 participated in boot camp and 82 did not
- 1 boot camp student and 11 non boot camp students have left (failed) the course at the one-third point
- 38 boot camp and 64 non boot camp students took the survey

This optional student survey is intended to help me know you better as a class and also for statistical records on the performance of different types of students (e.g., by intended major) in this class. The goal is to help us better teach this, and other, courses. All questions are optional.

1. Name: ____
2. Sex (please circle one): Female Male
    - 22 males and 16 females in boot camp, 50 males and 14 females not in boot camp
3. Age: ___________
    - Mean age of 23.5 years for boot camp, 21.4 years for not in boot camp
4. Year (please circle one): Freshman Sophomore Junior Senior Other (_________)
    - 5 freshman, 11 sophomore, 15 junior, 5senior, and 2 other for in boot camp and 22reshman, 18 sophomore, 29 junior, 8 senior, and 3 other for not in boot camp
5. Intended major: ______________________________________
    - 0 ChE, 4 CE, 22 CSE, 5 EE, 0 IE, 6 ME and 1 other for in boot camp and 0 ChE, 11 CE, 40 CSE, 4 EE, 0 IE, and 7 ME for not in boot camp
6. Programming experience (please circle one): Never Little Some Much
    - 15 never, 16 little, 5 some, and 2 much for in boot camp and 15 never, 33 little, 11 some, and 3 much for not in boot camp
7. Anticipated grade in *Computer Tools for Engineers* (please circle one): A B C D F
    - 25 A, 11 B, 2 C, 0 D, and 0 F for in boot camp and 53 A, 11 B, 0 C, 0 D, and 0 F for not in boot camp
8. Are you repeating *Computer Tools for Engineers* (please circle one): Yes No
    - No data recorded for this question
9. Current GPA at USF: ____________
    - 3.21/4.00 for in boot camp and 3.21/4.00 for not in boot camp
10. Are you a Community College transfer student (please circle one): Yes No
    - 16 transfer students and 22 non transfer students in boot camp and 27 transfer students and 34 non transfer students for not in boot camp
11. If "Yes" for (10), Community College GPA at time of transfer: ___________
    - 3.27/4.00 for transfer students in boot camp and 3.25 for transfer students not in boot camp
12. Course load this semester: __________
    - 12.1 credit hours for in boot camp and 13.2 credit hours for not in boot camp
13. External work or athletics commitment in hours per week: ________
    - 16.6 committed hours for in boot camp and 23.0 committed hours for not in boot camp
14. Is there anything else you think we should know about you? ____________________
    - No data recorded for this question

Ken Christensen

Ken Christensen received his Ph.D. in Electrical and Computer Engineering from North Carolina State University in 1991. He is currently an Assistant Professor at the University of South Florida. His research and teaching interests are in the areas of computer network systems, architectures, and performance modeling with an emphasis on integrating voice, video, and data in existing and future networks. He has over fifteen conference and journal publications, eleven U.S. patents, and several patents pending all in the areas of computer networks and performance modeling. He was awarded a USF 1996/1997 Outstanding Undergraduate Teaching Award and a 1998 State of Florida Teaching Incentive Program award. In 1998 and 1999 he was awarded a NASA/ASEE summer faculty fellowship at Kennedy Space Center. In 1999 he was awarded a CAREER grant from the NSF. Ken is a member of ASEE and ACM and a senior member of IEEE. His homepage is at *http://www.csee.usf.edu/~christen*.

Darrel Davis

Darrel Davis is presently completing his Masters degree in Computer Science at the University of South Florida as an international student from Belize. Darrel's interest is in computer science education and upon graduation, he plans to return home to pursue a career in the education field. This will be Darrel's first publication. He is a student member of ASEE and his homepage is at *http://www.eng.usf.edu/~dadavis*.

Dewey Rundus

Dewey Rundus received his Ph.D. in Experimental Psychology from Stanford University in 1971 and his M.S. in Computer Engineering from the University of South Florida in 1985. He is currently an Associate Professor and Associate Chair in the Department of Computer Science and Engineering at the University of South Florida. His research interests focus on Human / Computer Interaction and System Usability. His concern for undergraduate education has earned him a University of South Florida Outstanding Teaching Award and a State of Florida Teaching Incentive Program Award. His homepage is at *http://www.csee.usf.edu/~rundus*.