8-13-2018

Topp, William Oral History Interview

Doris Meyer

FACULTY EMERITI INTERVIEWS
UNIVERSITY OF THE PACIFIC ARCHIVES

William Topp (1970-2014)
Professor of Computer of Mathematics and Computer Science

August 13, 2018

By Doris Meyer

Transcription by Savannah Dughi, University of the Pacific,
Department of Special Collections, Library

Subjects: Math Department in the 70's changes emphasis - theoretical to applied; Department responses to student demands and corporate need for computer training; Computer science and computer engineering degree programs established by the University; Department of Computer Science with fully accredited degree programs established by the University; Computer enrollment and programs flourish in the boom period of the 90's; Computer faculty and programs become part of the newly formed School of Engineering and Computer Science; The years faculty members Bill Topp and Bill Ford published assembly language and data structures books that presented syntax and applications for emerging programming languages.

Meyer: Hello Bill.

Topp: Hi Doris.

Meyer: Here we go. Bill the first thing that we need to do is to say what we are doing, who we are, what the date is, and so forth. So I'll start and say Bill I'm Doris Meyer and I'm your interviewer. And who are you?

Topp: I'm Bill Topp, William is my formal name but I've always gone by Bill. I'm the interviewee.

Meyer: And we are in the library in one of the little study rooms and it's August 13, 2018. Bill tells me that this is a special day. How come Bill?

Topp: My wife Joan and I have been married forty eight years today. We will celebrate but for now Doris, this interview has priority.

Meyer: Okay and thanks for making arrangements so that we could meet here today. Bill and I have been involved with the Emeriti Society and the oral histories for some time. He knows that it is most important to get some information recorded on the important programs and other kinds of historical things that make the University of Pacific what it is. So Bill how did you arrive in Stockton California and at the University of the Pacific?

Topp: After I finished my doctorate in Mathematics at the University of Washington Seattle in 1968, I went back to my hometown Milwaukee and taught at Marquette University for a couple years. There I met my future wife and we decided that we should find a more pleasant place to settle. Experiencing cold snowy winters and short summers had much to do with this decision. I had some experience with California and she lived for years in Portland Oregon and so we decided to come west. Joan and I had plans to be married in August 1970 in Portland. At the national math conference I looked for a job with a university on the west coast. I interviewed with Bill Ritter, a mathematics professor at UOP and former instructor at the University of Washington. I had a campus visit and was hired.

Meyer: What did you think about things out here in Stockton California?

Topp: Joan and I were married in August 1970 in Portland and immediately came to Stockton. I only knew the city from my campus visit. Joan had a period of about four or five years when she lived in Sacramento and would come with her dad to see UOP basketball games at the Auditorium. Our shared experiences with California made us want to come west and Pacific was a beautiful small campus - exactly what I wanted. Joan was able to get a teaching position she enjoyed at Stockton Middle School, a junior high for the Catholic Diocese. I really liked my colleagues; Floyd Helton was a very understanding chair. We were able to make some friends and we soon took to Stockton. We were able to have a child relatively early in our marriage and that added a lot. My wife then began teaching part time at St. Mary's High School in the drama program and as the drama director.

Meyer: In what program?

Topp: Drama was her major and she loved directing a fall drama and a spring musical each year. This continued for twenty years. As our son was growing up, I coached his baseball and soccer

teams. These activities allowed us to meet a lot of very friendly people. We loved the diversity of Stockton and our lives were very active.

Meyer: What was the Mathematics Department like when you first arrived?

Topp: I joined the department in September 1970 along with three other relatively new PhD's. We were in the College of the Pacific housed in the old Quonset Buildings in what is now the front lawn of the DeRosa University Center. We were a department of 6 with Bill Ritter and chair Floyd Helton the one senior member. It was an exciting time. Floyd Helton put up with us young guys who wanted to change everything overnight.

Meyer: Can you remember the other three young fellows?

Topp: Yes. There was Bill Brown, Doug Smith, and a statistician. Doug and I were graduate students together at the University of Washington; Bill came from Dartmouth.

Meyer: What were the courses that you were responsible for in the beginning?

Topp: Well my field was Algebraic Field Theory and Ring Theory which are topics in a graduate program. The material was not in the undergraduate curriculum and so I basically taught the introductory calculus courses and linear algebra. A little aside - Floyd Helton told me on day one that I would be teaching a pre-calculus course in Raymond College. I had no idea about the cluster colleges or about non-letter grading. But what a delightful experience! I had a number of excellent students who subsequently came to COP after Raymond closed.

Meyer: Was mathematics part of the general education program? Or were your students primarily majors and what were they planning to do?

Topp: The Math Department was primarily involved in service courses. Engineers took a two-year sequence of calculus courses, statistics was required by the social sciences, and students preparing to be elementary school teachers took a specially designed math course. At Pacific, we had only a small number of majors and their curriculum covered the classical undergraduate math courses. In the seventies math departments nationally were experiencing a drop in majors as students saw that better jobs were in engineering and the sciences which used more applied mathematics.

Meyer: Bill can you remember some special people on the University campus and some others off campus that made that transition for you welcoming?

Topp: **Y**ou know there are so many people you meet as colleagues; but two stick out as special in my early years at Pacific. One was Cliff Hand. He was at COP when I arrived and later became Academic Vice President. A more gracious person you could not find. He loved what a good general education would provide students, a value I tried to give my students. He was a terrific fellow. And another one was Marc Jantzen. We met in the early 70's when Marc was Dean of the School of Education. At the time, the government in Sacramento wanted to make big changes in the training of elementary teachers. The Ryan Bill told schools of education that their curriculum had to greatly increase liberal arts requirements. Our School of Education needed to have a representative from the liberal arts. I was teaching the Math for Elementary Teachers course and was chosen. I went with Marc to a number of Northern California

education meetings. My role was to represent the liberal arts in these discussions. I gained a great deal of respect for Marc and the faculty in the School of Education.

Meyer: I played a similar role for credentialing in physical education. Let's move ahead and talk about student interest in pursuing a math major at Pacific. You indicated that students wanted the emphasis to be on applied math. How did that go?

Topp: Doris this was a difficult time. All of us young faculty saw the writing on the wall and wanted to see changes. We wanted to find ways to make our math curriculum more relevant for the day. Floyd Helton gave us free rein.

A year or so after I joined the department, we had a couple of openings. Even with new hires, we would continue to be a small department with a heavy service load. Trying to move our curriculum in a new direction was challenging, a little like turning a cruise ship at sea. We focused our hiring priorities on finding faculty members with strong applied mathematics background. We hired Bill Ford who came from the University of Illinois with a degree emphasis in differential equations, important for engineering and physics. We also hired Dave Hughes, a statistician with good experience as an industry consultant and advisor for graduate student projects requiring statistics.

At this time, the Math Department offered only a B.A. degree with concentrations in the traditional math areas (we called this the "pure math" option) and a probability/statistics option. The University was encouraging cross-disciplinary study and so we offered a new Cross-Disciplinary B.S. in Applied Math-Physics. We added a course in numerical analysis which emphasizes ways to find numerical solutions to problems that arise in the natural sciences, social sciences, engineering, and business. The solutions involve using computers for the calculations.

Meyer: Was this an attempt to start up computer science?

Topp: Not at all; that startup was still in the future. At this time there was a good deal of computing being done on large (mainframe) systems. The programming language FORTRAN (Formula Translation) gave engineers and scientists computer routines to carry out calculations that were previously done by hand (remember the slide rule). Another language COBOL (Common Business Oriented Language) allowed a computer to manage inventory and process payroll. Statisticians used SPSS Statistical Package for the Social Sciences) for data analysis and correlations. All of this was used by professionals and taught in university engineering and statistics classes. At Pacific, we had a Burroughs B6700 mainframe that supported a large lab where students could work on assignments and projects. There was computing on campus but this was not computer science; however, it certainly planted the seeds.

In 1974, we did offer an Introduction to Computer Science course. It provided students a chance to study a little FORTRAN, to learn how to communicate with a computer, and to run computer routines and packages. Our new math hires had some experience with computing. Bill Ford developed a background from his MIT undergraduate program and Dave Hughes made extensive us of available statistical packages.

Meyer: How did the new emphasis on applied mathematics proceed? Did it have the effect you hoped for?

Topp: To some extent yes. Roland diFranco became chair in 1975, a job he held for 3 years. A good number of courses in the curriculum needed to be updated to add an applied math focus. Whenever possible, a new course or independent study was added. The new emphasis was geared to stimulate student interest and use good marketing to increase enrollment. One example was operations research which is a branch of applied mathematics that applies analytical methods to help industry make better management decisions. I spent a semester at Stanford taking courses in the area. A colleague, Bill Brown, and I offered an Introduction to Operations Research evening course. It was well received but we did not have time or resources to really develop this area. Bill Brown continued to offer the course for several years.

In 1977, the math department began offering a B.S. degree and divided the curriculum into areas of concentrations which had designated faculty to teach the courses and serve as advisors. The areas were Pure Math, Applied Analysis (used by engineers, physicists, and hard science researchers), Operations Research, and Probability/Statistics. This was the way mathematics presented itself to students going forward. There was a concerted effort to support each of these areas. Admissions helped out with marketing but with only limited success. To the students it was still a degree in mathematics and they saw excitement and good jobs on the horizon if they took up computer science. At the same time, the department's service load was increasing. The social sciences required statistics, business had students take a finite math course, and additional calculus resources were needed to serve health sciences programs.

Doris, there was a very humorous side to our department's marketing strategy. It involved attaching a number to a course. For years, long before us young faculty came, math courses had numbers seemingly unrelated to one another - Oh yes calculus I and II would have successive numbers. But then our faculty took to creating a numbering scheme each time catalog revision was due. One year we decided that courses taught every year should have an odd number, every other year an even number. So the class Elementary Functions (pre-calculus) might go from 70 to 71. After we introduced areas of concentration numbers were changed so courses in an area would have their own block. Analysis got the 50's (51, 53, ..., 155) with upper division in the hundreds. Elementary Functions, a prerequisite for Calculus I (51), was numbered below 51 into the 40's block. It became 41. Within a few years, a course could be numbered 70, then 71, and then 41. Each year we went to Courses and Standard for approval of number changes. The committee gave approval but with disbelief that mathematicians would so struggle with numbers.

Later, when computer science courses were added to the curriculum, they needed their own block, a point of some contention. If numbered in the 20's they would appear to be less difficult than courses with a 30 or 40 number. I could go on but you get the point. It was a self-imposed dilemma.

Meyer: Not seeing a significant improvement in enrollment must have been disappointing. It took a lot of effort to carefully assess your curriculum, get the new courses approved, and revise catalog copy to market the program and emphasis. Where did you go from there?

Topp: Before we continue, let me tell you about an important event.

The Mathematics Department moved from the Quonsets to the South Campus around 1976. I remember somewhat fondly the Quonsets, old World War II metal buildings. Two of the offices had outside windows; the others were little dark-paneled caves. The department office was in an annex having a door from the chair's office and a door from the outside. To pick us one's mail, you hoped Stockton was not having a pelting winter rain. It could be a drenching experience.

The move was welcomed. Pacific had recently purchased the old Delta Campus soon after the junior college move to new facilities on Pacific Avenue. The campus had a few permanent buildings and a few high ceiling steel buildings for tech training and the maintenance shops (paint, print, automotive etc.). The University immediately referred to the Delta purchase as the South Campus. The Pacific campus was termed the North Campus or the Main Campus. The most important permanent building was the Classroom Building. That is where we moved. We were one of the first departments in the building; a few biology people were upstairs and a few chemistry people had labs on the first floor next to us. The building was pretty much unrenovated with a lot of empty rooms. As the Math Department grew on the South Campus we would commandeer rooms. Only a few in the administration knew what we were doing. In the summer, we obtained paint that was left over from Main Campus projects and, with minimum pay and plenty of pizza, we got some high school to create usable classrooms. We no longer had to trek up to Wendell Phillips or Morris Chapel for classes. As time went by, we expanded within the Classroom Building and made it a true home. I will leave the details for a little later.

Meyer: I remember that time. For us on the North Campus, you were "down there". We knew it housed some science people but few other details.

Topp: Doris, before I mentioned the move to the South Campus, you ask how the department was attempting to grow its major. There is a bright side here. In 1975, the catalog had an interesting statement "the University supports student interest in computer science." This was a low-key way to say the University recognized a ground swell of student interest in computers and would be on board (details would come in the future).

Many of our students were TA's at the Computer Center Lab which now is the location for HR. While they learned to help students with their course-related software packages, the TA's also learned how to access and use some of the features of the B6700 mainframe. Many wanted to go much further and figure out how to write code that would use its more technical features. One of our exceptional students, Don Gregory, was the poster boy for this. He understood he would need to learn ALGOL (Algorithmic Language), a language that was used by Burroughs in the design and implementation of its operating system. The language was developed in the 60's by researchers who wanted syntax and constructs that would allow a programmer to design algorithms and code much like a person would think. It was the precursor of most languages that followed. Don studied ALGOL with a vengeance and ended up publishing a detailed manual soon after graduation.

I mention Don and ALGOL because they were part of our computer science beginnings at Pacific. He and others wanted the University (our department) to provide the promised support. The problem was that none of our faculty was formally trained in computer science; Bill Ford had some undergraduate training while at MIT and became the leader in our introducing a few computer science courses into the curriculum. He offered an Introduction to Computer Science course using ALGOL. He also oversaw some independent studies for highly motivated students like Don Gregory.

As time went on, the department continued to offer the introductory course and added an applied computer class "Computer Approaches to Mathematical Problems". Bill Ford offered a new course that looked at data structures which looked at ways to enter, access, and update data as well as providing searching and sorting algorithms. This course, entitled Programming Languages, also introduced students to the theory of structured programming. The intent was to show students that "programming is a science not a bag of tricks." With these courses we had a skeletal computer science program.

There was a lead up to our selecting these courses and writing catalog copy. In the 60's a few larger universities had computer science programs. They were often in schools with a strong engineering emphasis like Purdue University that established the first computer science degree in the early 60's. What these early programs should contain and be described was an issue. The ACM (Association of Computing Machinery) published recommendations for academic programs in computer science. The report was termed Curriculum '68 and helped develop computer science as an academic discipline and gave a framework to describe courses and programs. When use this report to help us get started.

Meyer: Bill, it's amazing how computer science kind of crept into the department course by course. Bill Ford was the lead, how did you become involved?

Topp: In about 1976 I took on the pleasant task of exploring computer science. At night, I had a computer monitor and modem on a table near a phone. The modem worked by inserting the phone handset into the modem and calling the University Computer Center. An annoying problem was that I had a wonderful dog who would every so often bark causing the modem to disconnect. I was frequently redialing. I started out teaching myself FORTRAN. I learned the process of entering a program, compiling it, and creating output. Having to find syntax and logic errors (debugging a program) was a great learning tool. FORTRAN is easy for a mathematician since a program would often look like a series of math equations used to solve a program. Also engineers and scientists had created a good deal of FORTRAN code to solve familiar math and physics problems. I learned a lot about how to design and code routines by tracing existing quality FORTRAN code.

After a short time, I turned my attention to learning ALGOL and a more in depth study of CANDE (Command and Edit) which was the Burroughs user interface. It had instructions to compile and execute programs, and manage file systems.

Meyer: You just described how you got involved with computer science. How did this carry over to the curriculum?

Topp: Doris, I think a little background is needed. In 1977 the math degree became a B.S. (Bachelor of Science) degree. I think the B.A. in mathematics was retained requiring fewer courses. It was clear the department needed to serve the growing number of existing math majors who were interested in a formal study of computer science. Admissions saw that a number of students that wanted to come to Pacific expressed an interest in computer science. We had students from other programs that wanted to switch majors and begin studying computer science.

In response, the department added computer science as an area of concentration that could be used to meet degree requirements. The department understood that that it had to build a more robust computer curriculum that gave students exposure to a good range of topics, had introductory programming courses, and established prerequisite requirements. Bill Ford was assigned full time as a computer science instructor.

It was clear that Bill needed support. I agreed to help and began to work closely with him. We became very good friends, actually lifelong friends. We interacted with each other throughout our careers at Pacific. I taught the introductory computer science class using the BASIC programming language. ALGOL was introduced in the Data Structures and Programming Language course. Topics like compiler construction and operating systems were offered as independent studies. The department introduced a "1st Course in Numerical Analysis" which looks at numerical approximation techniques that use computer algorithms. It covers important everyday problems in math, computing, engineering, and physical science disciplines. It gave our students experience with often-used applied computer science. It was later adopted by Engineering as a required course. The skeleton computer curriculum continued to grow.

Meyer: Let me get this straight. Your "computer science students" were still identified as math majors getting a B.S. in mathematics with a concentration in computer science. Their program involved a good number of upper division computer science classes. But they were not computer science majors, right?

Topp: Doris, it was a little confusing. The administration understood this. The University was about to make important changes that would create Computer Science and Computer Engineering degrees. Trying to keep this all straight is difficult. I hope I can recall important ideas and put them in some kind of time line. Good lord, it was some 40 years ago.

A little background is important. In the 70's there certainly was a strong interest in computing but the field of computer science was very young. There was a feeling that computer science as a field of study should simply be part of electrical engineering or mathematics. In fact, a good deal of early computer training was part of the curriculum for these two disciplines. Over time, as computers were used for more and more tasks, it became clear that designing and writing programs for computers to solve problems was difficult and requiring theories and practice unlike those used in existing fields. Computer science needed to be recognized as a discipline in its own right.

We had great demand and interest in computer science but no separate degree or major. Seeing what was happening at other universities, our administration took action. It was understood that there were enough resources in math and electrical engineering (computer hardware) to begin offering Computer Science and Computer Engineering degrees.

Academic Vice President Alistair McCrone understood this. He saw the great demand and interest in computer science and that Pacific had no separate degree or major. Seeing what was happening at other universities, he took action, recognizing that there were enough resources in the mathematics and electrical engineering departments to get started. He pulled together faculty from these two departments and charged them to create computer science degree programs which he termed "separate but complementary." He understood that some new hires would be required but wanted the programs to share as many courses as possible. Administration recognized that the hires and the equipping of a computer lab would be an expensive proposition. The pool of potential hires was small and they expected high salaries. Computer hardware for a lab was very expensive - a small 10 megabyte disk cost thousands of dollars. Today you can get 100 times that much for pocket change!

A 6-person task force was created, 3 from mathematics and 3 from electrical engineering. Bill Ford and I along with Roland diFranco came from math; the engineering group included chair Dale Dunmire and an experienced faculty member with good computer background, Ron Pullyblank. I had just become chair of the Mathematics Department.

Meyer: What happened?

Topp: We met and created the computer science and computer engineering degree programs. The process to organize curricula so the programs could share courses, to get University approval, and to make the hires took about a year. The math department began offering the Computer Science degree in 1979 while engineering started theirs in 1980. In the planning stages we envisioned a lab that would serve equally both degree programs. We moved ahead with this goal in mind. Things ended up quite differently.

Meyer: Was this lab going to be in the Computer Center?

Topp: Doris, the story of the lab has many interesting twists and turns. The planning group knew that the lab should serve the new programs in a separate facility. Basic computing needs would continue to be handled by the Burroughs and the Computer Center. We got the idea that it would be nice to have the lab situated halfway between the math department on the South Campus and computer engineering in Baun Hall. We opted for the Wendell Phillips Building and acquired a small classroom on the second floor. All this was a bad idea. Without proper air conditioning for the terminals, the small room became stifling hot especially when it was jammed with students. It became clear we needed to find a larger space.

At the same time, the planning group came to understand that the lab would primarily serve computer science which would need to design the lab, purchase the equipment, and then manage it. Computer engineering would need labs that catered to hardware. Colleagues from other universities gave us good advice. They made it clear that if we wanted to start a program like this, we needed to make sure that students had their own computer system; otherwise they would go in and crack every computer on campus. We already had experience with this. We settled on a large room in the basement of the Classroom Building and looked into getting a

separate computer. The arrangements would work fine once the room got a coat of paint and new ceiling lights.

Equipping the lab was important. We purchased a PDP-11 minicomputer from Digital Equipment Corporation (DEC). A best seller in the 70's and 80's, it had hardware features and a UNIX-based operating system that became a model for others. At this time AT&T made available to universities the documentation for UNIX. Boxes containing reams of manuals were shipped to us. They were very well used.

Bill Ford downloaded and then modified a software system developed at the Newcastle University (England). The system, termed the Newcastle Connection, was designed to interconnect UNIX-like systems. Bill provided students a network of terminals, disk drives, and printers, all connected to the PDP-11. The lab was a hotbed of learning. Students used the software and documentation provided by AT&T and Newcastle to gain networking and systems programming skills. This was important as it helped students find ready employment in Silicon Valley.

Topp: Doris, recalling the lab has me thinking about some crazy things that happened while getting computing growing in the department. I hope this might give the reader a little better feel for our goings on. With the lab in the basement, the department wanted to connect faculty offices to the PDP with access to the Burroughs. It was critical for faculty teaching computer classes so they could conveniently work with students. With little regard for our health, Bill Ford and I went up into the ceiling crawl space of the Classroom Building and climbed around heating ducts to drag a cluster of wires designated for the different offices. The crawl space and ducts were covered with dust - God knows what else was there! Proudly we had a primitive computer network.

Meyer: You indicated that the Computer Science degree was approved. Did this degree become part of the math department offerings and what happened?

Topp: Yes. At this time, computer science is within the math department. Bill Ford and I went to the ACM (Association of Computing Machinery) conference and were fortunate to find an excellent candidate who had gotten his degree at Rutgers and wanted to come west to be with family.

Our computer curriculum expanded, in part from our cooperation with electrical engineering. We were able to offer a set of courses that presented important topics in computer science - systems programming, operating systems, compiler construction while engineering offered a microcomputer course. Mathematics introduced a course in discrete structures which looks at topics like combinatorics, graph theory, and sets/logic. Computers are critical for solving problems in these areas.

There was a significant change in our computer curriculum when we began using the language Pascal in lower division courses. Pascal is a computer programming language developed by Niklaus Wirth of Switzerland to teach structured programming. It is a robust language with relatively straight forward syntax and structure ideal for teaching students. It was adopted by most universities for their introductory computer classes. Bill Ford and I flew to the University of California San Diego in July to get a newly written Pascal compiler that would run on the

Burroughs. August was used to learn the quirks of Pascal and its implementation on the Burroughs. I introduced a new course "Introduction to Computing with Pascal" in the fall with a hand-on lab. The students enjoyed the course and so did I, despite the fact that I often had to stay up late night writing lectures and testing of the software. Beginning in 1979, my teaching load became the offering of computer science classes. It represented my fulltime transition to computer science.

One more thing: the University was encouraging students to look for co-ops in the summer. The Office of Cooperative Education existed in COP and we looked to it for guidance. More importantly, we got help from engineering which required two semesters of co-op in its 5-year programs. A number of our students got co-ops which were supervised by the co-op employer and the math department.

Meyer: No doubt it was important that students could say they got a degree in computer science rather than a degree in mathematics with a concentration in computer science. You were chair now. Could you just settle in and relax?

Topp: To some extent. We had a good set of computer courses in our curriculum and students were becoming more comfortable with the mix of computer science and computer engineer courses. With some adjustment, our students did pretty well in engineering labs that dealt with testing circuits and wiring boards.

In 1981, we hired Ken Hunter, who had a master's degree in computer information systems. This is an area that develops computer models and applications for industry and business management Ken may have lacked the PhD but he had a good deal of consulting experience. He was a great colleague and totally committed to helping students. He was a wonderful hire.

As more and more students were looking for the computer science degree, we could identify different areas of interest. Some were your techy-type who took computer classes and then hung out in the lab. Others saw that a strong math background would appeal to many employers and would open opportunities. Still others wanted less of the systems training and more of computer applications, particularly for business. In the catalog we presented different options for the major - Systems, Mathematics, and Information Systems. The majority chose Systems which required taking most of the computer classes. The Mathematics option required a solid computer background but left space for students to take upper division applied math courses.
This group often took overloads so they could complete both the Systems and Mathematics options. I had a couple of very bright and motivated students who took a number of physics courses in addition. They are having wonderful careers at Lawrence Livermore Sandia Lab.

With Ken Hunter on staff, we could offer a information systems concentration that had students taking fewer computer systems course but adding courses in economics and business management. Most of these students wanted to end up in business but with a knowledge and appreciation of a computers' ability to process data, simulate production models, and manage

inventory. Initially there was some reticence by computer science and School of Business students to choose this option. They felt it was a computer degree for the "less capable". Ken dispelled this by recounting his experiences and presenting exciting and challenging information systems classes.

Meyer: You said things were pretty stable and comfortable, but only "to some extent". What were some of the issues that were challenging?

Topp: Doris, to put it simply it was enrollment, enrollment, enrollment. I was still chair for mathematics and for computer science and continued so through 1984.

For mathematics, the number of traditional math majors stayed pretty constant. It was enrollment in the math service courses that grew. Statistics was being required by more departments, the growing pre-med and pre-dent programs required calculus, and enrollment in engineering remained strong. The University had a general education program that had a math competency requirement. This involved creating the Math Resource Center that served those with limited high school math background. Class sizes got too large and faculty teaching loads were almost exclusively service courses. When I met a faculty member to discuss scheduling, class sizes in low level service courses always came up. When the topic came up in department meetings, I smiled, thank everyone profusely for their commitments and promised I would try to get help. The COP dean was sympathetic but totally unresponsive.

Enrollment for computer science had its own dynamic. Remember this was a time when PC's became popular. Businesses had an IBM PC on each desk and kids were enjoying their new Commodore 64. Silicon Valley startups were popping up seemingly everywhere. President Reagan greatly increased the defense budget which had a trickle-down effect on hiring for a whole range of defense related companies. Interest in being linked to a computer education was important in the minds of many students. An example will tell the story. Each year, Admissions would have an on-campus open house for perspective students. In one session we had 2 new perspective math majors and two classrooms in Wendell Phillips filled with interested computer majors. I ran out of brochures and chatted well into the afternoon with parents. Students on campus would daily come in for advising and getting papers to change majors.

Staffing courses was a nightmare. We had a long list of upper-division courses that needed to be taught each year. New majors needed the lower division courses which were also popular with students around the campus that wanted to explore computer science. The programming courses needed labs for hands-on experience and so class size had to be limited. We needed to schedule multiple sections for these courses.

Bill Ford and Ken Hunter were fulltime teaching computer classes. This was not enough and I began to need (to commandeer) faculty from the math side of the department to teach some of the lower-level computer science classes. Dave Hughes was an obvious candidate. He was very familiar with computers from his work with statistical packages. Dave was a good teacher and had access to notes and sample programs I wrote when teaching the introductory programming class. We had one or two math faculty that volunteered to help out. Doug Smith was the exception. His graduate studies in math logic were good background for his transitioning into our computer science group. He was enthusiastic and capable. Doug developed a senior level

course in computing theory that, amongst other things, looked at Turing machines which provided a mathematical model of computation. Doris, you may have seen the genius of Alan Turing portrayed in the movie "The Imitation Game" where he developed an actual version of his machine that broke the German's Enigma Code in World War II.

Recruiting and retaining computer science faculty was always difficult. I have already noted that the pool was relatively small although the number increased gradually as universities added graduate computer science programs. You had to compete with the needs of industry and the appeal of working with a research group. Salary was a big issue. We needed to hire new people at a salary that was significantly higher than our existing math faculty and most others in COP. Our administration in the early 80's did not help. A suddenly imposed blanket freeze on hiring and discussion that tenure would be curtailed led to Ken Hunter leaving. Computer science was attracting students and building a good reputation. We needed to add trained faculty.

Meyer: You said you were chair of the Mathematics Department until 1984. During that time you were overseeing separate math and computer science degrees. Was it not the case that your faculty members identified as being in one group or another? This must have been a little awkward. Describe how things developed.

Topp: Doris, I felt a growing split in the faculty. We were in one department with one chair but some were "computer people" others were "math people." It was difficult for some of the math faculty to see all the enthusiasm and innovation that was happing with computer science while they were asked to teach mostly service courses. As chair, I tried hard to service both math and computer science. Colleagues were sympathetic and realized that computer science needed the bulk of my administrative time. The situation was understood by my secretary and office staff. I ended my time as chair proud of what the entire department was able to accomplish and I was very grateful for the support provided by my colleagues and Dean Benedetti.

Meyer: After your term ended, did the new chair experience the same problems?

Topp: Yes, my 6 year term ended in 1984. I was pleased that Dave Hughes became chair. He was well liked and I thought the department should have a chair chosen from the math faculty. While Dave was responsible for the two quite separate degree programs, he let the computer science group set up the curriculum and schedule classes and teaching assignments. He got permissions for us to hire but let us do the recruiting. This arrangement continued for a year or so until we became a separate department with a separate chair.

Meyer: By the mid-80's, you had built a strong computer science degree program with good enrollment. How did you move forward?

Topp: At that time we were offering an information systems option within the computer science degree. However by offering only a few information systems courses and merely having students take economics and business courses, we were not adequately exposing them to the field. Students in computer information systems (CIS) would need an understanding of business management models and decision making processes. Business, communication, and organizational skills would be needed. Internships with businesses that made extensive use of computers throughout the company would be important. In many universities, substantial

investments in CIS programs were made and the degree was promoted. A growing number of graduate schools began offering a PhD in information systems.

In 1985 we hired Kouros Mohit who had recently graduated with a PhD in CIS. We asked him to look for ways to develop the information systems option within our program. He introduced into the curriculum Communication Systems Analysis, Information Systems Management, and Decision Support Systems. These courses, together with our database management course and our lower-division computer science offerings described a solid information systems option.

A year after Kouros was hired, we began offering a B.S. in Computer Information Systems. This gave our students a choice of which computer degree they wanted to pursue. Some came to Pacific wanting the CIS degree. However, in many cases, a student was studying computer science with knowing the long-term direction they wanted to take. Once they got into the program, students would assess their level of interest in the more theoretical side of computer science or in applications, specifically for business. Advisors discussed the options with a student when laying out his or her course of studies. A good number chose to pursue the information systems degree. They were well aware of expanding job opportunities. There was more than just a career Intel, Cisco Systems or same gaming company When companies like Amazon and Google came along, good jobs in information technology, communications, and marketing opened up. The same opportunities became available at local wineries or agricultural companies like Foster Farms.

Doris, two of my advises graduated with the CIS degree. They used their enterprise skills to build multi-million dollar businesses.

Meyer: You mentioned that Dave Hughes was your chair for on a couple of years. What changes came about?

Topp: In 1987, the Department of Computer Science was established. Doug Smith became its chair. There is a good deal of irony here. We started out as a single department with one degree occupying one space in the South Campus Classroom building. We then were a single department with two degrees in the same space. Now we are two departments with two chairs in one space. There were obvious administrative issues. Two separate department meetings were held, office staff now had split responsibilities, and the two chairs would often have quite different discussion with the dean when hiring and salary issues were involved.

Around this time I recall the university encouraging departments and schools to offer minors. It was hoped that students would build in a minor to promote cross-disciplinary study. Our department offered both a major and a minor in its Computer Science and in its CIS degree programs. The minor required students to take fewer computer courses combined with a pared down concentration.

On the academic front, the University wanted to insure that its graduates would have an ability to write. This is hard to define for different disciplines. In computer science one might think

that writing logically correct programs would be a measure of writing skills - our students wanted to think that. Our grads got a rude of wakening when they were assigned the job of writing documentation or a guide for a new software application. Our friends in engineering took a lead for their civil, mechanical, and computer engineering students by offering a course in technical writing and we gladly made this a requirement for our majors.

Meyer: Having two different departments in the same space using the same office resources is unusual. How did this all resolve?

Topp: In the mid 80's, biology and chemistry enrollment was booming with strong student interest in healthcare professions. It became clear that these programs needed more space in the Classroom Building especially for faculty research labs. Mathematics would need only a small section of the building provided computer science faculty and programs could be relocated. We were instructed to move to Hand Hall on the North Campus. More irony here; we started as the math department in the Quonsets on North Campus, moved to the Classroom Building on the South Campus and now was returning to the North Campus as the Computer Science Department. Doris, later you will discover that we were not done moving!

Meyer: What was the Hand Hall move like?

Topp: We moved into the first and third floor. I recall that the Communications Department was on the second floor. The third floor had been remodeled to provide business faculty with offices before they moved to Weber Hall. It housed our offices. We held department meetings in the chair's suite which was a long narrow room distinguished for its many slanting roof beams (duck you head!).

On the first floor we set up two labs, one a general purpose lab for our computer service courses and the other for our majors. A small machine room held a computer server, storage drives, and the hub for a local network which we set up to connect faculty offices and the labs to the server. Today we think of wireless communication. In those days network communication was done by dragging insulated copper wire from "pillar to post". UOP employed two skilled AT&T telephone technicians tasked to set up a campus-wide telephone network. We needed their telephone equipment and technology to wire an 8' by 4' panel that was the hub. They became great friends and volunteered to make the connections. They thought we were a little crazy (which we were) but God bless them.

Connecting the faculty offices to the ground floor server was a challenges. Fortunately, dragging wires from the third floor to the machine room was possible only because Hand Hall had originally been a student dorm. It had a shaft (for laundry?) that ran down through the floors to the basement. A couple of days into the move, we carefully ran 3 wires from each faculty office along the third floor's drop-down ceiling. Very carefully we put tags with a name and a number at each end of the wire and sent them down the shaft to the machine room. Our AT&T friends found it interesting that they were relying on a paper tag to make a hub connection. This was a low tech operation in many ways.

Meyer: Once you moved into Hand Hall how did things go?

Topp: Once we got used to walking 3 flights of stairs to get anywhere, the new surroundings worked well. It was nice to have offices that were away from the noise of students moving from classroom to classroom in the Classroom Building.

Accreditation became an important task. During the 80's, almost every university began introducing computer science programs with different curriculum and emphasis. As time went on, universities felt it advisable to put some structure and quality controls onto the various programs. Rather strict accreditation guidelines were established to setup minimum curriculum and data gathering requirements to insure goals were being met. With the computer field changing so rapidly, it was important that students and employers have confidence in the graduates being produced. Our professional computer organization ACM (Association of Computing Machinery) set up a Computer Science Accreditation Commission (CSAC) which established the guidelines for accreditation.

We initially did not go for accreditation because we did not have the faculty in place and had only recently become a separate department. While we had a good track record for producing computer science grads, our future viability required that we get accredited. Without it, we would have been viewed as a second tier school in the eyes of prospective students.

As I recall, we began in 1989 to seriously plan for an accreditation visit. We needed some adjustments in the curriculum and also needed to gather records on student performance, graduation rate, employer reviews, etc. Doug Smith did an excellent job preparing documents. We submitted an application in 1990 and were granted a limited 3 year accreditation in 1991. This was typical of CSAC for first time applicants. No accreditation was then available for CIS programs. We made sure our program followed the recommendations of the Data Processing Management Association (DPMA).

Meyer: You got accredited for only three years. What did you do about getting the full six year accreditation?

Topp: Doug Smith became a site visitor for CSAC. During visit to several universities, he got an insider's view of how the accreditation approval process works. In 1993, I again became chair, this time chair of the Computer Science Department. I was happy that Doug took charge of our preparing documents for the next accreditation visit at which time we got a full 6 year accreditation.

My work as chair was also helped by new hires for Computer Engineering. This complementary program was greatly enhanced when Ken Hughes and Louise Stark, a husband - wife team, joined the faculty. Their degrees were in computer vision and artificial intelligence; Ken had training in robotics. Our students liked their courses which helped satisfy the networking and digital systems requirements for accreditation. We developed close relationships with the two.

Meyer: I recall that the 90's featured a real tech boom. How did all this affect computer science at Pacific?

Topp: I was chair from 1993 to 1999. During that time program was blessed and at the same time overburdened with the number of computer science majors. We had almost 100 majors which were near the highest total in COP. The 90's was a pivotal decade for computing. Many

nowadays just take for granted the tools, applications, and hardware that drive the marvels of modern computing. A lot of it had origins in the 90's. Intel and Motorola produced 32-bit processors that made possible high speed computing on a PC. Microsoft introduced Windows 3 which made available a graphical user interface on a PC that resembled the one available on the expensive Apple Macintosh.

Growth and use of the internet flourished in the 90's. It had its origins as a network developed by the Department of Defense in the 60's to link computers at Pentagon-funded research institutions over telephone lines. In the 90's the internet found many commercial applications particularly when the World Wide Web came along. This featured a browser that permitted hyperlinking of web pages one to another and to downloadable files, documents, and other web resources. The internet became a powerful communication medium that had an impressive number of uses and users. Google was just an upstart company that used artificial intelligence to revolutionize searching for information on the internet. The corporate giant Amazon was founded but did not really grow until the 2000's. One could go on and on. It was an amazing time. From high school on, kids wanted to be involved with computers and many chose computer science as their major.

Keeping up with the enrollment demand was difficult. The administration was hesitant to approve new faculty positions and attracting quality new hires was a challenge to say the least. There was not a sufficient pool of graduates with PhD computer degrees and those that had the degree were drawn to industry with very high salaries. We were attracting, teaching, and graduating a wonderful group of students each year. They joined video game companies, internet and system security teams at Lawrence Livermore Labs, and startup companies throughout Silicon Valley. It was the best of times for our program.

Meyer: Computer Science was doing well. I assume the same is true for Computer Engineering. How was your relationship with engineering during the period?

Topp: Both of our programs were doing well. You seem interested in our relationship with the engineering faculty as a whole and not just with those involved in the computer programs. This is a great question and lets me mention Ash Brown, the School of Engineering dean during this time.

Ash came to Pacific from Detroit with background in the automotive field. From this experience, he saw that there should be close cooperation between computer science and the different fields of engineering. Each year, Ash organized a 3 day retreat at Feather River for his entire school. It brought together the different engineering disciplines and promoted a good camaraderie. He made sure that we were invited and that we had an active role in the agenda. It was a great setting for us to get to know the faculty from engineering. Sometimes Ash arranged to have a consultant that would bridge computer science and engineering programs. Besides growing friendships, the retreats enhanced cooperation which carried over after Ash retired and Ravi Jain became the new engineering dean. I don’t think Ash could appreciate the value of having us at the retreats. We had good times together.

Meyer: You know Bill, it would be interesting to read the transcript of Ravi Jain's oral history with Ash Brown the interviewer and also Ash Brown's oral history with interviewer Bob Benedetti. You had a good chance to appreciate them from the perspective of being a chair.

Topp: Doris, Ash was certainly a positive force. But there is someone else, Lee Fennell, who played an important role in the growth of the computer science program.

University procedures made it difficult for computer science to purchase and maintain computer equipment. Unlike computer engineering that had designated budgets for each of its program, our COP dean, Bob Benedetti, designated line items in the budget for each department determined by its traditional needs. For computer science, that meant faculty salaries. He had no budget for computer hardware and was told our program should rely on university resources. Fortunately for our sake Lee Fennell, the former registrar and now Acting Academic Vice President was given the added task of being our liaison with administration when we needed to negotiate for new equipment.. We knew Lee, liked and respected him, and found him very willing to work with us. But Lee was not given a budget. He would have to get us funds on an item by item basis.

Our reliance on Lee grew in the late 80's and early 90's when the administration was looking for ways to cut costs. We had a Computer Center staffed by University personnel. However when it was decided to purchase Banner, an administrative software application developed specifically for higher education institutions, the company was contracted to bring in its own staff and run the Center. The arrangement catered to administrative needs with little support, financial or personal, for academic computing. Lee understood the situation and did everything he could to help us. Unfortunately, Lee was never given the credit he deserved.

Meyer: I am glad you brought up Lee. I think we all share your feelings. So let's get back to your department. You were still in COP, our liberal arts school. How did that work out?

Topp: There were issues. We believed in the mission of COP and wanted our students to have a good level of liberal arts exposure but degree requirements that met accreditation guidelines imposed limits. The department was not able to contribute to the college's liberal arts program which was resented by some. Salaries became the real issue. Dean Benedetti understood the problem and let us hire at competitive rates and then tried to somewhat equalize salaries for our existing department faculty. Computer Science had a pay structure that was out of line with the rest of the College. I think the writing was on the wall. Computer Science should move into one of the professional schools, most properly the School of Engineering.

Meyer: You described how the 90's saw the birth and then expansion of internet-web uses, the launching of new companies, and development of new areas of application. How did you adjust your curriculum to keep up with all these advances?

Topp: It was a little wild. I remember having to rewrite my notes each time I taught the same course. There are a couple of developments that are worth noting. In the beginning universities wanted students to become familiar with a "body of knowledge" that was computer science - computer systems, programming languages, operating systems, and so forth. As time went on, it became clear that the graduates would need skills to learn quickly on their own how so as to use the

new computer research and tools. It was not good enough to simply learn the technologies of the day, they would be gone or superseded tomorrow.

We introduced a course on Application Development which required that a student design and complete a project. Emphasis was on team-building throughout the design, implementation, and testing phases. Industry demanded that a new employee be able to work in a team which would stimulate creative ideas and provide the manpower for the different elements of a project. This approach found its way into our courses up and down the curriculum.

In the 90's, users took quickly to the visual appeal of an operating system like Windows and browsers that made the Web so attractive. We introduced a course in GUI (Graphical User Interface) programming where students were taught how to use dropdown menus and graphical elements like buttons and icons. Communication is performed by interacting with these elements rather than the usual text-based or command-based communication. GUI elements are now in almost every application.

In the late 90's, universities began to look seriously at the quality rather than simply the quantity of learning the computer science students were getting. Doris, this involved introducing features that you and I look for in a liberal education. They became guidelines that future accreditation teams would use in assessing a program. Let me describe briefly a few of these features. Of course it was assumed the curriculum would include basic computer science theory and topics. But graduates would need to deal with complex software applications that require teams of developers. There is an organized (scientific) process to build these applications which is referred to as Software Engineering. The study became its own course and was used throughout the curriculum. Other qualitative aspects of computer science training was learning communication and team work skills and becoming conscious of the social and ethical issues that computers present.

Meyer: You mentioned that moving the department out of COP was inevitable. Did this occur during the boom of the late 90's?

Topp: It was a little later. The department had to go through another migration while still in the College. Around 2000, President DeRosa was embarking on a building / renovation plan for the campus. Some involved relocating services into different buildings on different parts of the campus. He was tearing down the Quonsets to make room for the Student Center, health science programs were to be housed on the Brookside campus north of the Calaveras, etc. Our move involved the South Campus - more irony. We started on the North Campus (Quonsets), moved to the South Campus (Classroom Building), then back to the North Campus (Hand Hall), and now south again. Old timers like me recall that Delta College had shops on the back side of the campus for painting, carpentry, and automotive repair. These shops were for campus maintenance and students wanting tech training. The university had already remodeled a few of these shops to house the Art Department and Geology Department.

The president wanted to relocate some administrative offices into Hand Hall. Computer Science was told to move into the old Delta College print shop that was now serving as the university's duplicating center. Part of the space was assigned to engineering for a new lab. The print shop

was a cavernous open space high ceiling metal building. We had a chance to map out every square inch of the space, designing labs, computer classrooms, faculty offices, and the department office. Circumstances put me in charge of the project. I would be reporting to Dean Benedetti.

Meyer: It seems strange. Doug Smith was then chair of the department.

Topp: Doris you are right. Doug had become chair in 1999. In his second or third year as chair he had a fall sabbatical leave and asked me to be the stand-in chair. As fall approached Doug suggested that since I would be involved in plans and scheduling effecting the entire year, I should be chair for the entire year. It was then that renovations occurred and the department moved.

I worked very closely with Tom Bowe who was the architect and with Financial Vice-President Pat Cavanaugh. I cannot tell you how much I appreciated Pat. Budgets were tight; nevertheless, when Tom Bowe and I would come to Pat with requests, he listened and gave us thoughtful and honest yes or no responses. I wanted to create a department facility that would serve us well for years into the future. I wanted it to be as creative and as futuristic as possible. I really enjoyed the project. I love working with detailed spacial relationships and Tom Bowe was great at taking ideas and sketching out the construction details. I had the aid of my colleague Mike Doherty who was skilled at creating 3-D computer representations of offices, rooms, and desk layouts. I wanted to have good sized offices that would allow faculty to have their computer on a separate desk with room to help students. The plans allowed for the outer offices to have good window lighting and for the inner offices to have special sound proofing to block out lector and lab noises. My colleagues had some excellent ideas. Space was limited but they wanted to have a large lecture room (with computers on each table), a lab for our majors with tables and whiteboards for team projects. There was also to be a special research lab. Initially it was not clear how this special lab would be used. As time went on, we acquired projection panels and computer equipment for dynamic wall displays and the showing of student projects. It would help us explore new computing equipment that today is common place.

Meyer: Then what happened?

Topp: Doris, as a matter of record, computer science stayed on the South Campus for 10 years, until new facilities became available in the engineering complex. But this is all part of a bigger story. The School of Engineering got a new dean, Ravi Jain, who came to Pacific from a university that housed computer science within engineering. Ravi was insightful. He wanted the same for his school at Pacific. He saw the potential for enrollment and just as importantly wanted many of the new ideas that were part of computer science to become part of his engineering students' training. Our faculty was pleased. Ravi saw that computer science salaries were not competitive with colleagues at other schools. He promised to right the situation - he was true to his word. Ravi developed a good relationship with Dean Benedetti that was built on trust and respect. The two deans set in motion our move into the School of Engineering.

Meyer: Computer Science was a department in COP. What happened when you became a department in engineering?

Topp: We officially became part of the School of Engineering in 2002. There was an adjustment period. Some faculty and students in engineering were amazed that they had not heard of the move. Ravi did everything to welcome us and place our faculty on committees. There was a drawback. We had to deal with physical separation. Computer Science was a department in engineering which was located on the North Campus while were a good deal away on the South Campus.

Perhaps, more importantly, computer science was a four-year program while engineering was a five-year program with two semesters of coop required. Computer science students did not have the same coop requirement. Our ACM accreditation was held one year sooner than engineering's ABET accreditation.

In the beginning, we wanted to retain our old policies and procedures. This led some engineering faculty to see us as still a COP department. Soon, we instituted changes that had us better fit in with engineering policies - with the exception of coop.

Our faculty had built up personal friendships with many in the engineering school, especially those teaching in the computer engineering program. After a year or so, we felt comfortable in our new school. Ravi was a good promoter. He established that we were now the School of Engineering and Computer Science (SOECS). The large plate glass window on the stairway to his office was beautifully embossed with the University of the Pacific Emblem and "School of Engineering & Computer Science."

Meyer: What was going on with computer science after you moved to SOECS?

Topp: There we a number of changes. In a nutshell we moved to the North Campus and important advances in computer technology led to new emphases in the curriculum. Early on enrollment suffered somewhat from the tech bust of the early 2000's; In 2004, Bill Ford took over as chair. He was good at assessing department and program needs and getting help from colleagues. Bill stayed on as chair until 2014, when he retired.

Ravi Jain wanted to modernize facilities in the engineering school and have faculty and administrative offices in one building. He wanted to build state-of-the-art labs. After almost a decade in fund raising, he proudly oversaw construction of the John T. Chambers Technology Center in the engineering complex facing out on Dave Brubeck Way. The center featured among other things a CNC ("computer numerical control machining") lab and a digital learning lab. It had the latest in environmental friendly features and achieved LEED Gold Certification. Ravi moved the computer science faculty from the South Campus to the new building. Part time faculty and TA's remained on the South Campus.

In recent times, computers have been involved in big technology developments and areas of application. Amongst these are cyber security (controls to protect systems, networks, data, etc.), artificial intelligence (systems with intellectual qualities like reasoning, discovering meaning, etc. typically found in humans), and data analytics (science of analyzing raw data to make conclusions about that information), and software engineering. We looked for hires that had special (degree) training in these areas. By combining strong theoretical computer science training with focus on new areas of computer applications, Pacific could market the B.S. in

Computer Science to students. Employers liked the product. The major had almost 200 majors and good graduation rates.

Meyer: Computer Information Systems (CIS) was an important area of study, a concentration, and ultimately a major in your department. With the move to engineering, what happened with CIS?

Topp: A growing number of graduate schools began offering PhD's in CIS. Being only an intermediate size university, it was difficult for us to allocate sufficient resources to build a robust program. For our students pursuing the CIS degree, we did require them to take economics and business courses. But there was no real buy-in from these disciplines. Those courses were designed explicitly for their programs and continued to be taught that way. Only indirectly did our students see links to their CIS degree.

When I was chair for a second time, I had hoped to stimulate a strong interest in information systems among our faculty and with key people in business/economics. Conditions weren't right and I was not persuasive enough. To build a strong CIS program, the provost and dean from business would have needed to establish a task force that identified critical links between business and information systems. New CIS hires would be need to teaching in the program and in business. Once computer science moved to engineering, emphasis was placed on systems programming and technical applications. CIS was not adequately served while in engineering. I think the program could have found a home in the School of Business and flourished. None of this happened. After we first moved to engineering, the CIS degree was maintained for a while. But with the tremendous growth in computer science, resources were needed for that program and CIS was dropped as a separate degree. This still pains me because there is projected to be a very high and growing need for information systems graduates as the computing becomes more and more part of business operations.

Meyer: It is difficult for a university our size to make the financial commitment necessary to create new program. Look how much planning and discussion went into developing the School of Health Sciences and expanding parts of it into the Sacramento campus. We talked about the move and the curriculum changes after coming to engineering. What was happening with you personally?

Topp: I went half-time starting in 2005 with a full teaching load but with no advising and reduced administrative duties. I formerly retired in 2009, but stayed on for 5 years as a part-time instructor. Each semester I taught one or two sections of Applied Numerical Methods using Matlab. The course was designed to give engineering students computing tools for their course work and future employment. It was required of all engineers except those in computer engineering and electrical engineering. I wrote a complete set of lectures notes with handouts so that I could cover most of the applied numerical methods that my students would need. I enjoyed the continued contact with them and felt the course material was very relevant.

I stay in contact with colleagues and marvel at the work they are doing. After the country came out from under the economic woes of the housing collapse, computer science enrollment has grown exponentially. Now computer science has hundreds of majors and is the largest department in SOECS. It future is very bright.

Meyer: This interview has involved mostly tracing the history reflecting the creation and growth of a computer science at Pacific. But you and I have spent a good deal of time working together as members of the Pacific Emeriti Society board. During this time, you showed me a copy of a book you and Bill Ford wrote that was translated into Chinese. I recognized your names on the title page and almost nothing else - it made for a good laugh. As part of this interview, I would like to get some idea how your book publishing came about and what you did.

Topp: The opportunity and discipline to publish computer science books was important for my career at Pacific. I began this interview by saying that I came from a theoretical branch of mathematics and, by a program of self-teaching, moved into the field of computer science. The publishing provided important depth to my background and my ability to offer relevant (and hopefully interesting) lectures. My colleague Bill Ford and I were joint authors for the books. We didn't want to write one book and then push out edition after edition with only minor updates. Rather we chose to have each publication be a new book featuring new computer developments. The first book was in assembly language for the Motorola 68000 family of processors. It was published in 1988 a year after we began writing, doing the research, and learning the language and whims of publishers.

Meyer: Go ahead, but first please explain to me what assembly language is?

Topp: Some background is required. Computer science students must learn about the internal architecture of a computer and how it operates (its runtime system). This involves studying the integrated circuit (processor) that has hardwired instructions to perform basic arithmetic and logic operations and others to move data and respond to signals within the system. For the runtime system, a programmer uses binary code (sequences of 0's and 1's) to store data and carry out instructions. In early computing, the coding was done manually - a very abstract and tedious process, prone to error.

Assembly languages were created to make more natural the programming task. Processor instructions were identified by mnemonics such as ADD (arithmetic operation), MOV (transfer data to or from memory) and TST (test a signal). Assembly language programs were written as a sequence of mnemonics and then translated into the binary code that directs the running of the program.

Bill Ford and I taught our assembly language course. In those days, we could have designed the course for an Intel processor (IBM PC) or for a Motorola processor (Apple Macintosh). We felt the Motorola 68000 processor had a cleaner design and would be more understandable. We looked at available textbooks and concluded we could do a better job and set out to find a publisher. In 1988, the MC68000 Assembly Language and Systems Programming book was published. The first nine chapters were material for a one-semester course providing a detailed look at the architecture of a computer system and a formal presentation of the Motorola assembly language. The last five chapters looked at advanced topics that emphasized applications of assembly language programming to systems programming which requires a good degree of hardware awareness.

Doris, all of our books concluded with design and implementation of algorithms for advanced applications. We wanted students to have this material as references in later courses.

Meyer: You said working with publishers was interesting - how so.

Topp: The editor is vital to a project. In this case she got excellent reviewers who provided timely, frank and helpful criticisms. Bill Ford and I wanted to include every last Motorola instruction with a detailed explanation and example. We went overboard. Our editor cajoled us by making clear she was interest in publishing a textbook and not a telephone book.

In the writing of a book, the author and publisher have different goals. One wants to create a manuscript and the other wants to plan its printing. Early on, I roughed out how I thought the book should look. The editor took the ideas and immediately set them in stone. We were encouraged (cajoled) to submit manuscript copy that conformed to a format. I spent many an early morning hour with the compositor squaring our submitted pages with the guidelines. In the end, Bill and I were proud of our work. Motorola purchased a good number of copies for its training courses.

Meyer: You said you did not want to publish editions with only minor updates. Were there other editions?

Topp: Yes, but I don't want to get too detailed. Motorola created a new processor that had significant changes. In particular, it added hardwired instructions specifically designed for structures and algorithms that were being introduced in new programming languages. To present the new processor, Bill Ford and I wrote the book "Assembly Language and Systems Programming for the M68000 Family." For marketing sake, it was called a Second Edition but was in reality a rather complete rewrite. When published in 1992, we took pride that every Motorola instruction was presented in detail with documentation and an example.

To complement the publishing of the assembly language books, Bill Ford and I wanted to write a user-friendly software package that would help students write, assemble, and run an assembly language program. In the process, students would see how data is handled by the runtime system. We wanted students to understand that a computer is not a "black box". We called the software package MAS (Motorola Assembly System). Neither Bill nor I had every written a computer program of this size (several thousand lines) and complexity. The software had a number of integrated components - a customized editor, an assembler, and drop-down menu commands to call features of the runtime environment. We provided a library of routines for user input and output in decimal, hex (base 16), and binary (base 2) format. We provided a debugger which allowed a student to step through a program one instruction at a time. It was useful when trying to identify incorrect code and analyze how a program "flows".

MAS was a project that was truly a labor of love. Writing it was personally invaluable to us as computer scientists. More importantly it promoted student learning and their enjoyment of the assembly language course. The small version of MAS came out for the first book. An updated version was created for the new book including a 110 page guide with page after page of screen captures. Today the guide would be presented in a YouTube video!

Meyer: You have talked about your assembly language publishing. You indicated before something about data structures. What is all this about?

Topp: In the 90's Bill Ford and I turned our attention to the field of data structures. The topics are fundamental to a student's computer science education and are used extensively by programmers. It makes sense. In a way you can think of a computer as a data storage and data handling machine. A data structure provides a data storage format that allows for efficient data access and updates. Doris, I am sure you are familiar with a number of different data structures - an ordered list (numbered bank checks), a first-come first-serve queue (waiting line at a pharmacy), a table that stores data in rows and columns (spreadsheet), etc. Applications that use data structures are literally everywhere.

What interested Bill Ford and I was looking at new programming languages that facilitated using data structures in applications. In each of our data structures books, we begin with details of the language, then covered the general topics of data structures, and concluded with applications that effectively used the features of the language.

Our first book was entitled "Data Structures Using C++" published in 1996. This programming language is significant for two reasons. Number one, it is an extension of the language C which was developed by AT&T computer scientists to create the UNIX operating system. You may not recognize the name UNIX, but you will recognize the name DOS which is mini-UNIX written by Bill Gates for PC's. It launched Microsoft. For decades, UNIX has been the world's most recognized operating system. A study of a system like UNIX and knowledge of C is a key element in a student's computer studies.

A second reason C++ is important is that it featured object-oriented programming constructs which are used today in the design and implementation of most software applications. It changed the face of programming. Doris, I think you will understand the idea of an object with an example. A bank has savings accounts. The definition of an object with a name like "savingAcct" could be a way for a computer program to view an account. Now include in the object's definition relevant data and operations. The data might include the account owner, its current balance, and a list of transactions since the last statement, etc. Relevant operations might be "deposit", "withdrawal", "processAutoPay", etc. The object "savingsAcct" is an abstraction, a way a person (and programmer) thinks about a savings account. Doris, for you and I the bank has separate instances of this object. The computer might identify your account with the name "savingsAcct. meyerAcct" and my account with the name "savingsAcct.toppAcct". An object is a customized (programmer defined) data structure with a toolbox of supporting operations. An object-oriented language like C++ allows for a more natural way to viewing a problem and creating a solution. By bundling data and operations, it provides a programmer constructs that result in better, faster, and more user friendly applications.

Meyer: This gives an idea of the content in your books. I am interested in the Chinese book.

Topp: Doris, the Chinese book is a by-product of our second data structures book which presented new ways program designers were expanding C++ constructs. In data structures, the focus was initially on implementation issues so that programs could efficiently store and access large data sets within the limited physical resources of the computer system. As computers developed greater CPU power with increased memory and storage capabilities, researchers and practitioners were free to give more consideration to the abstract design of the data structures.

In 1993 researchers proposed that a data structure be viewed abstractly as a bucket (a container) that holds data with operations to add, update, or remove items in the bucket. Some containers store data in sequential order, first item, second item, etc. Other containers associate a separate tag (key) for each item. A container like this is used by a package delivery company like UPS or FedEx with a tracking number as the key. An abstract operation like "remove" makes sense for a sequential storage container and for an associative storage container. A programmer would like to simply use the command "remove" knowing what it means (an item is deleted) without having to worry how the operation is carried out.

Well, this programming desire was addressed at a C++ standards meeting in 1993 where a researcher presented a library (code syntax and routines) that defined different containers with their operations. The library, called STL (Standard Template Library), was enthusiastically received and is actively used today. In 2002, Bill Ford and I published the book "Data Structures with C++ Using STL" and used it in the data structures course to show students how new resources and ideas are coming on-board to use data structures in applications. Doris, this is the book that Prentice Hall had translated into Chinese. We knew it was read because suddenly we had students from China emailing us for help in solving some of the exercises.

Meyer: You were getting ready to go half-time. Did that want to ease off on publishing?

Topp: The language Java was a popular programming tool for developing computer applications. The language was developed in the early 90's using C and C++ syntax with a goal that computer people could produce "write once, run anywhere (WORA)" programs. It required having a compiler that would take the Java code and translate it to run on a Windows machine, another compiler to translate the same Java code to run on an Apple, an Android, etc. machine. Today, Java is open-source software that runs on everything from laptops to data centers, game consoles, and scientific supercomputers.

Bill Ford and I published the book "Data Structures with JAVA" in 2005. We used the introductory chapters to present the syntax and features of a new JAVA language which we made available as a primer in the appendix. JAVA includes and advances many of the ideas found in STL. In the latter chapters we applications for graph theory, data compression, and so forth which are tailor made for JAVA.

Meyer: Bill, we are getting near the end. If we had not done an interview like this, what would have been lost? I mean aren't you proud to take a panoramic view of your career that mirrors so closely the evolving years of computer science at Pacific.

Topp: Doris, you have been the inspiration and driving force behind the Oral History Project promoted by the Emeriti Society. I read some of the interviews but personally doing one has been very satisfying. I hope anyone who reads it will feel my pride in teaching and advising students who would make a career in the computer field. Doris, I want to thank you and the library staff that made the interview and its publishing possible.