Title
Using personal computers in introductory programming classes : an appraisal of a management decision

Authors
Feldman, Julian
Volper, Dennis

Publication Date
1986-05-14

Peer reviewed

Using Personal Computers in Introductory Programming Classes-- An Appraisal of a Management Decision

Julian Feldman and Dennis Volper

Department of Information and Computer Science
University of California, Irvine
Irvine, California 92717

Technical Report 87-33

14 May 1986

Using Personal Computers in
Introductory Programming Classes–
An Appraisal of a Management Decision

Julian Feldman
Dennis Volper

14 May 1986

1. Introduction

During the Winter Quarter of 1983, the Department of Information and Computer Science at the University of California, Irvine, decided that something had to be done about the provision of computing services for its three-course freshmen sequence in programming and problem solving.

Over the previous years the number of students enrolled in these courses had grown dramatically from 604 in 1977-78 to 1325 in 1981-82 for an average of 441 per quarter, and Fall 1982 enrollments were 30% higher than Fall 1981 enrollments.

Unfortunately, the computing equipment available for instruction during this period did not increase at the same rate as enrollment. The failure to provide adequate equipment was caused by the tight budget of this period, underestimation of enrollments, and a lack of consensus on appropriate equipment for instruction.

Two obvious manifestations of the shortage of computing resources were students waiting in line to use terminals and very extensive use of computing resources for instruction during the third shift (midnight to 8:00 a.m.) and on weekends. Two other consequences were the increase in the amount of cheating detected by instructors in these classes and the increase in complaints about the adequacy of computing resources from students and from their parents.

The Department had three alternatives: reduce assignments utilizing computing resources in these three courses, reduce enrollments in these courses, provide additional computing capacity. The faculty would not accept alternative one. The department chair did not think he could convince the campus administration to accept alternative two. And so the department began to explore alternative three.

2. The Crisis in More Detail.

Not enough hardware capacity. At first glance, the problem appeared to be solely the result of not having enough computing capacity to handle the student load. However, when we began to examine the difficulties in this three-course sequence, we found at least three other problems.

Three versions of Pascal and three environments. Although the faculty had decided to standardize on Pascal as the language of instruction in this three-course freshmen sequence, we did not have the same version of Pascal and the same programming environment for all three courses. During the early '80's we were using UCSD Pascal on single-user LSI-11 based systems in the first course, another version of Pascal in batch mode on the DECsystem 10 for the second course, and still another version of Pascal on the SIGMA 7 for some offerings of the second course and the third course. To add to the confusion, we acquired additional single-user systems from another vendor to provide additional resources for the first course; so students in the first course were divided into two different groups. The poor freshman's problems—using three different dialects of Pascal and three programming environments (operating systems, editors, file systems)—were compounded by an academic calendar consisting of three ten-week quarters in which there is precious little time to get sick or even to get confused.

Daytime staff and nighttime students. As we indicated above, the inadequate computing resources forced students into heavy usage of computing during the off shifts and weekends. Virtually all of the faculty and teaching staff available to help students were on duty during the daylight hours of the standard Monday-Friday work week. So students who had problems during the second or third shift or on weekends had to ask other students for help, tough it out (increasing computer usage), or give up (increasing frustration). We should not have been surprised to see an increase in cheating.

Cost. The 1982-83 university budget was still tight; so we were faced with providing substantially more computing resources without substantially increasing the budget. On shared computing resources (DECsystem 10, SIGMA 7) UCI had a fairly standard charging policy based on usage. The problems with these policies have been discussed at length elsewhere, and we shall not dwell on them here. But even though we were spending about $100K per year for these three courses—about $60 per student per quarter; we never seemed to have enough computing. Even when the machines had idle time, the students had to use second and third shift time to conserve budgets.

On the single-user micros there was no "charging" per se. The allocation policy was essentially first-come, first-served with some sign-up facilities and scheduling, but the number of such systems was so small with respect to the demand that they were used 16-24 hours per day.

Peak-loading problems. We also encountered severe peak-loading problems at assignment due dates and at the end of quarters. The peak-loading problems were caused by a shortage of resources, human nature, and the self-paced structure of some of these courses.

3. Our requirements.

Connect hours. We wanted enough capacity to provide 500 students with six connect (terminal) hours per week during sixty daylight hours (six day per week, ten hours per day). So we would need at least 50 stations.

Scheduling paradigm. To improve the quality of computing time, we decided to adopt a different model of student access to computing. The model we had been using is a "library" model. Computing resources were provided, and the student decided when to come to the resources and use them. The terminal room (microcomputer lab) was staffed with a proctor whenever the room was open. The proctor's main duty was to protect the equipment; the proctor was not always knowledgeable about the course material. Educational staff (instructors, teaching assistants, tutors) were available during normal working hours, but they were not always in the same room as the computing equipment.

The model we adopted is a laboratory model based on the model used in the laboratory sciences where students have a fixed schedule of hours in a laboratory room. One class at a time is in the lab, and at these hours appropriate instructional staff members are in the lab. During scheduled laboratory sessions, the student is guaranteed access to computing resources. Conversely, the student will not have access to computing resources during at those times he or she is not scheduled. Because classes were scheduled in the lab for only 10 hours per day, six days per week, we could make the remaining hours available on a first come-first served basis, i.e., a library model. We had some modest debate over the use of the unscheduled time. On one hand, it would be a pity to waste the time and leave the resources idle; on the other hand, instructors might not tailor assignments to the length of the lab periods if they knew that the standby capacity was available for slower students.

Hardware/Software. We wanted to provide a uniform Pascal environment for all three courses. Because some instructional material for the first course was written in UCSD Pascal, we decided to use UCSD Pascal or a compatible version of Pascal. We also wanted an easy-to-use editor; and we wanted a file system which would permit students to merge programs, i.e., incorporate tools written early in the quarter into larger programs written later in the quarter. The system had to have a translator and sufficient hardware capacity to handle the edit-compile-execute-debug cycle which is the basic paradigm in programming classes. Students are not building production systems, but they are invoking this basic cycle 10-15 times per hour, and the system had to be able to provide good response to this kind of load. In addition, we needed a system which could respond to the peak load requirements which invariably occur in student computing. Usage peaks before due dates and toward the end of the quarter. While we hoped the laboratory model would smooth some of the peaks, we knew that scheduled labs could not remove all of them. We also wanted to provide each student with about a half megabyte of disc storage and we wanted to avoid some scheduling problems by choosing an environment in which a student would be able to use any station. Finally we wanted a high degree of availability. Our experience with old, balky time-shared systems convinced us that the new system should have a degree of availability which would not interfere with the pace of instruction in a ten-week course where even two or three days of down-time during the quarter can disturb the pace.

Summary. We wanted a system with UCSD Pascal or a compatible translator, an easy-to-use editor and file system, a fast edit-compile-execute cycle,i.e., able to handle 500 edit-compile-execute cycles per hours, the ability to handle moderately large programs (500 lines), good response under peak load, 50 simultaneous users, and 250 megabytes of storage. We also had some additional goals related to possible future activities—running Ada on this system, using the UNIX operating system, and simulating a multi-processor system. We hoped to accomplish all this on a budget of about $150K.

4. Alternatives.

We considered at least six classes of alternative hardware solutions:

(1) A single machine which would support 50-60 simultaneous users was one alternative. A DEC VAX 11/780 was a possible alternative, but it was a little weak on CPU for our needs. Other possibilities were a single processor Gould machine and a multi-processor Convergent Technology machine.

(2) Several superminis could also support our load. At the time we were making our decision, DEC was offering a package of five VAX 11/750's for $150,000 to customers whose utilization might benefit DEC. Other contenders in this category were Sun and Apollo.

(3) One faculty member was advocating supermicro class machines, e.g., WICATs, which might support 5-6 users on each system.

(4) A popular choice was a single-user machine based on a Motorola 68000. While advocates of this alternative conceded that the 68000 might be overkill in a single-user system for the immediate application in freshmen courses, they argued that future software would be more CPU intensive and that we would probably be getting 68000 based workstations for research and/or upper-division instruction.

(5) The single user system utilizing an Intel 8088, e.g., an IBM PC, was not the most popular choice initially. But IBM held out some hope of a gift of PC's, and many of us realized the advantage of the large amount of software for the IBM PC.

(6) We also talked to Apple and made them an offer in June of 1983—give us 50 Apple II or III's to use until you can deliver Macintoshes which we would buy at the consortium price. We also looked at DEC mini's.

5. **The Choice and How We Got There.**

(1) Our first choice would have been the five-pack of 750's. We already had two 750's, and compatibility would be very nice. One of us (DV) had implemented UCSD Pascal on a 750; so we had the software we wanted in hand. We were a bit nervous about the cost of filling our the 750 configurations (the DEC deal included only basic machines). And we were concerned about a file system broken up over five machines. But DEC wouldn't go for a five-pack deal with us—so that was that.

(2) One of us (JF) thought that the Convergent Technology system looked very nice—a combination of distributed computing and a shared file system. But CT did not have multi-processor software available at the time we had to make a decision—and the other of us (DV) kept us from jumping too soon. The proposed Gould configuration software was still being developed and looked too risky.

(3) Everybody was excited about the possibility of a gift of 50 PC's from IBM. And the IBM salesman gave us the impression that the gift was a possibility. But as the deadline approached, the IBM salesman retreated. Finally we made IBM an offer. They turned the offer down. But we continued to explore various IBM PC deals from vendors in southern California.

(4) The Apple III's didn't look like a good interim solution for us; Apple didn't want to include us in the Macintosh consortium; and Apple didn't want to loan us the Apple III's until we were able to get delivery on the Macintosh. So our conversations with Apple stopped.

(5) In the process of exploring sources for third party floppy disk drives for IBM PC's, we were referred to Corona Data Systems—a manufacturer of PC clones. After some discussions with Corona and their local distributor, they made us an offer which beat anything we could do on an IBM PC and gave us better graphics, a bigger power supply, more expansion slots, and a motherboard which could hold 512K bytes of RAM.

While we looked at a few other alternatives (e.g., NEC APC, Toshiba PC), a number of choices were not considered because the cost was nowhere near our budget or the software was nowhere near what we required. Thus, we only seriously considered the five alternatives listed above. Two of them (750's and Macintoshes) were eliminated because we couldn't make the deal that some of our peers were making; and we couldn't afford the list price. One (CT or Gould) was eliminated because the software wasn't ready. Stand alone 68000's were too expensive. The 5-6 user 68000's didn't seem like a good choice, that is, didn't dominate single user 8088 based systems and left us with unresolved questions about how such a system could be used with large classes. So it came down to a choice between IBM PC's and the Corona PC clone.

We chose the Corona clone because of price and possible future expansion capabilities. Because our first application called for a single language and we had people who knew the UCSD p-system, we weren't worried about software for this first application.

6. Getting Started.

Once the selection decision was made early in the Summer of '83, we faced a number of other problems in getting the laboratory ready for students in the Fall of '83.

Space. To obtain even a modest amount of laboratory space, we had to move a hardware lab, takeover a conference room, and move a dozen or so graduate students. This displacement took precious time and made us a number of enemies. Because the space was not available until late summer, we pushed back the delivery of the hardware. Then we got very nervous that we wouldn't get the hardware in time for the beginning of classes.

Wiring. Because the space had previously been used for other purposes, we had to make arrangements for adequate wiring for the personal computers.

Furniture. We wanted to make good use of the modest amount of space we had; we wanted to make the space reasonably attractive; we wanted the chairs to be adjustable; and we had a budget constraint. The vendor of our first choice for tables could not meet our delivery requirements; so we selected a local vendor who could meet our budget and delivery requirements. The tables have been quite satisfactory. The chairs were a bigger problem. We selected a good looking chair from a major manufacturer of stylish office equipment, but the chairs have not held up as well as the tables. Chairs take a lot more wear and tear than tables; and in a regular office environment, the chairs would probably be quite satisfactory. But in a student lab, we should have opted for strength rather than style.

Hardware. The projections of how we would use the system indicated that we would like two disk drives, for the additional storage. This storage included students files and resource files (compilers, test inputs, utility programs). We were faced with the choice of purchasing a second drive or filling RAM from 128K to 512K. These options provided nearly equal storage for nearly the same price. We chose the fill the RAM on all machines and get a few machines with dual drives to make disk-to-disk copies. We also purchased a couple of hard disk drives, which we have never used. We experienced a modest number of problems in testing the machines when they arrived. The vendor had obvious problems in burning in machines and testing them prior to shipment. Other problems surfaced when we started stuffing memory chips into the sockets on the mother board. A number of sockets had problems. While it is not reasonable to expect the vendor to check every socket on every motherboard, better quality control or acceptance testing on the sockets was certainly in order.

Software. After a modest debate about whether we should take the SofTech version of UCSD Pascal and modify it to run on the Coronas or buy the NCI version of the kernel and add on what we needed, we decided to go with the NCI version because it was faster. We had to do some programming because we chose to utilize the higher density Corona graphics as opposed to the IBM compatible graphics mode. Finally we had to make some changes to the course material because we were implementing a more recent version of UCSD Pascal than we had been using.

Security. We were concerned about the security of the equipment very early in the planning process because we did not plan on keeping the lab open 24 hours a day, seven days a week, and because of the portability of the equipment. The laboratory spanned a number of adjacent, interconnected rooms. The initial discussions were over how the locks should be keyed and who should have which keys. We also installed an alarm system which signalled a control panel in the police station (central plant) if a door was opened after the alarm was armed. So far the only detected unauthorized entry was by two students working for the police department who had master keys which permitted them to enter the lab when it was closed. Loss of computer equipment has not yet occurred, however, the chairs have a tendency to wander into adjacent classrooms.

Cost. Including the cost of the extra RAM the computers were just under $2K per machine. The principle expenses for setting up the space were for wiring, chairs and tables. Each of these items cost about $100 per station (total $300). To assure a high percentage of stations would be occupied by functional machines we purchased 10% spares. Setting up the software involved approximately 4 man months of work. We we went to buy spare parts, it turned out to be cheaper to buy additional systems and disassemble them as needed. Of the small expenses the most costly the were printers and the security locks; under $2000 for these items. In the final analysis the lab cost us about $2600 per station.

7. Management issues.

The weakest portion of our planning and implementation was management. We knew we needed to support both the instructional process and the hardware. We anticipated a need for separate staff for each of these functions. Even though we planned for a lab manager and a modest amount of "hardware" staff, we underestimated the workload required. The people assigned to this lab had other duties outside of the lab which competed for their time so the amount of time allocated to the lab varied. As a result certain non-emergency tasks, such as periodic cleaning of the labs and the machines, were neglected. We successfully provided instructional support in terms of tutors familiar with the course material during the laboratory hours, but we were unsuccessful in our attempt to provide support for the instructors themselves. We attempted to use a lab instructor for such things as lab exercise development, but we failed to integrate this person into the course staff. The software seemed to span the instructional and support staffs. Who should handle software such as print spoolers, editors, compilers; and who should handle computerized quizzes, and course book keeping systems? As a whole we underestimated the amount of maintenance support needed and its cost.

8. The Plus's and Minus's of our Selection.

The decision to use personal computers in our introductory programming classes really involved three separate decision: (1) The decision to use single-user personal computers; (2) The selection of particular hardware and software systems, i.e., Corona personal computers and UCSD Pascal; and (3) the decision to schedule laboratory periods.

(1) The decision to use single-user personal computers. The personal computers fulfilled our expectations in providing the large amount of computing service we required and the high level of availability we desired at a very modest hardware cost.

Previous experience with micros alerted us to the frequency of diskette and disk drive failures in a floppy disk environment. These did occur, however outside of the initial burnin period, the error rate has been tolerable. Most errors are sporadic (suggesting dust) and the overall rate corresponds to the length of time since the last cleaning of the lab.

Without a local area net or other connections between the PC's or between the PC's and a central server, we had to resort to brute force methods for system initialization, file transfers, and printing of files. While the lack of a local area net reduced hardware costs (and software complexity), it increased people costs and user costs by forcing users to move diskettes rather than ship files to printers, etc.

Some of these costs were overcome by a series of decisions and/or adjustments. We decided to configure so that the boot placed the system into RAM disk. The student's disk contained only personal files. This increased the speed of the system, decreased the number of disk accesses (thereby reducing disk errors) and avoided frequent reboots. The RAM disk was also used to distribute files to the students in lieu of a network.

The problem of printer access was severe until we realized that the speed of the printers themselves was the bottleneck. Additional printers were made available and some machines committed as print servers.

Despite considerable experience we understated support costs. So it came out of the hide of faculty or out of the hide of student users. Maybe we wanted PC's and control so much that we fudged on continuing costs. Facility costs are usually bundled and include such things as lab cleaning, software support and supervisory personnel. With micros the tendency is to equate hardware costs to total costs. However, the cost of supporting the laboratory operation (per enrolled student) seems to compare favorably with our cost of maintaining a more traditional configuration (VAX) for our upper division students. We also discovered that capability generates need. We now could supply additional hours per student and could tailor software to the educational need. Thus, the level of service we ended up providing to the students exceeded our original estimates.

(2) The decision to buy Corona's. Our original reasons—lower purchase costs, better graphics, cheaper memory expansion—were fine. But the other reasons we had—bigger power supply and more expansion slots—turned out to be moot because we haven't used them.

The principle complaint about the Corona's were "they were not IBM's". We had some initial problems with the Pascal system. These were in three categories. First, a slightly different boot sequence hung the entire process. Second, the IBM BIOS is proprietary and cannot be copied. Third, we decided to use the higher density graphics rather than the IBM compatible ones.

The first two problems were mostly solved after we vigorously prodded the software and hardware vendors. The third (along with part of the first two) was solved by hiring a graduate student to produce the modified software.

Although the above solutions made our decision highly effective educationally the incompatibility had other effects. The most noticeable was the decision of the department not to adopt the Corona as the standard departmental administrative machine.

Maintenance of the machines has been a variegated experience. The decision to do first-line maintenance in house has been hindered by turn-over of critical staff but in general has been good. As mentioned above, there were some burn in problems with the machines, but these were short term. The decision to have spare systems helped greatly. Our current procedure for machine failure is to swap out the bad machine for one of the spares so that the failure can be examined at leisure in the shop. The maintenance people have complained of unreliable and difficult to find memory problems, mostly with bent pins on the chips we stuffed, but statistically our problems have involved under 1% of the stuffed chips. Overall the reliability of the machines favorably compares to other micro computers we have used. On the whole, service has not been bad, however it would probably have been more convenient with IBM's.

The area in which many are worried that we will take the worst beating is resale value. Since we dream of upgrading the laboratory to better machines; this is of some concern. We would be much better off trying to sell IBM's than a compatible. However, the correct computation is total depreciation in value. The best for which we could resell IBM's is about $500 below the cost of a new one. The difference between this and the cost at which we could have purchased the IBM machines is not much less than the purchase cost of the Corona's. However, as far as the campus is concerned, second-hand IBM's would be snapped up by other departments, the Corona's probably will not.

All in all—a marginal decision.

(3) Laboratory model. The laboratory model reduced contention for computing resources and reduced the amount of time students spent waiting in line for access to computing resources. It made better use of staff time because we arranged to have students and staff there at the same time. The laboratory model —because it was based on a traditional academic laboratory model in which the students are provided with necessary access to the lab during (more or less) the daylight hours of the Monday-Friday work week— required additional capacity when compared to a traditional computing center model which assumes 24 hour a day, 7 day a week operation. In fact, the laboratory model helped us justify additional capacity.

Disadvantages. One of the disadvantages is that the laboratory model restricts student's schedule. While we like the idea of self-paced activities, we think that self-paced activities work better in a self-paced world. But in a mixed world where one class is self-paced and the other three classes are not, the self-paced class loses out. There is academic validity to March and Simon's version of Gresham's Law of Planning— unprogrammed activities lose out to programmed activities, so the scheduling did have some advantages. What it lacked was sufficient flexibility. For the largest class, this we have instituted a scheduling program to make empty seats available for students who want them. This alleviated the problem, but has not completely solved it.

What should we do when students miss a lab session? During the initial quarters we had considerable excess student demand for computer time. So we tried to fill in empty seats during the day, and all we did was generate queues. A similar problem developed when we attempted first-come, first-served time for evenings and weekends. Fortunately, as we processed the backlog of students waiting for these courses, the demand for time was reduced. The acquisition of some additional computers also helped; modular expandability did have its advantages.

Space problems. We compromised to get started, that is, we started with less space than we really needed. Averaging 20 square feet per station (counting space used for aisles, printers, administration) it was far too tight, and dedicated mostly to student occupied machines. There was little room was left for printers and administration. Movement was sometimes a problem in congested aisles. (And then students want to bring their bikes into the lab.) A slight increase (about 10%) in space and the establishment of some rules alleviated, but did not eliminate the problem.

The final problem pertained to faculty cooperation. The decision to shift to the laboratory model of computing needed to be accompanied by changes in the manner which students had to do assignments. There was not enough faculty effort on rewriting assignments to fit lab schedule. As a result the labs experienced peak loading problems, particularly as students seem to need 24 hours (or more) of computer time during the day before the assignment was due.

9. **Summary.**

Basically a good decision. The problems that we have encountered were caused mainly by not going far enough, e.g., interconnecting the PC's with a local area net, revising courses to match laboratory scheduling, providing necessary support. The support matter may be part of a scaling up problem. JF's parents started packaging nuts in cellophane bags around the dining room table. In estimating the cost of production, they failed to include the cost of rental space and the full cost of labor. So when the business expanded and they had to move into a storefront and hire non-family members to work, they discovered that their prices (based on earlier cost estimates) were too low. They had problems scaling up. Similarly, the level of support required by a faculty member may be less than that required by a naive freshman.

Was this the best possible set of decisions? We can't make that judgment, but the decision moved us to a better state of the world at a reasonable cost and it didn't cause any disasters. In retrospect, it was probably (overall) as good as anything else we could have done for the same bucks and better than many of our other options.

What contributed to making this a good decision?

(1) There were a number of people who were dedicated to making it work. These included faculty members, university staff engineering staff, vendors, purchasing, and students.

(2) The planning was pretty good. While some of the decisions were made a little late, we in fact anticipated most of the needs; and we were able to cope with the problems which arose.

(3) Few people were involved. Relatively few faculty and staff are associated with the courses which used the lab; and some of them were not present when fast decisions had to be made. So many of the decisions were made by a small group which was able to work together and which was willing to accept each others judgments. It is interesting to contrast this decision with our department's seeming inability to get something done on upper division computing resources. Part of the problem with the upper division decision is that many more faculty involved, and they are much less willing to compromise. Thus it is harder to get consensus.

(4) The problem space was small. One of the things that helped was we were looking for a solution to a very limited problem. We were only concerned with a small number of classes, with one programming language. We did not have to worry about a variety of users with a wide variety of software requirements.

(5) The preceeding crisis situation. Given the difficult situation which preceded the lab, almost anything we did could have been hailed as an improvement. As is we did better than "anything".

(6) Hardware reliability. After the initial burn-in problems, the hardware proved to be quite reliable. And this reliability contributed to the success of the effort by overcoming mistakes like inadequate support and problems like technicians leaving.

(7) Compromises. Another thing which allowed the project to fly was the willingness of the primary participants to accept compromises, e.g., space, start-up problems.

(8) Whatever happens may be for the best. We were concerned that these 8088 based systems would not be powerful enough to handle future plans, e.g., UNIX, Ada. But maybe it is just as well, i.e., maybe junking these machines after 4 or 5 years and moving on to something else may work out better than buying a more powerful system a few years ago. This will place fewer constraints on next step and the cost ratio has improved dramatically in the interim.