Constraints on Teaching in Computing Science

Professor E. S. Page

The teaching of software at the level that we have been discussing in the Seminar would normally be fitted into a specialist computer science curriculum whether at first or higher degree level. It would, accordingly, be subject to several constraints as are most degree courses in whatever subject in most universities. Some of the more important of these constraints are shown in Table 1 in the left hand column.

Table 1

Constraints

	Example B.Sc. (Honours) in Computing Science Newcastle Upon Tyne
1. Students	
a) Ability	Good
b) Preparation	1 Year : Mathematical
c) Intention	Specialist
d) Time	2 Years : Full-time
e) Number	6-25
2. Staff	
a) Number	14 Academics
b) Interests	Varied
c) Commitments	Considerable
3. Facilities	
a) Computer - Hands on	1130/KDF9
Closed Shop	KDF9/360 Model 67 (O.S.)
Terminals	360/67 + 2741's
b) Software	Varied
c) Ancillary Equipment	Some
d) Laboratory	Yes
4. Examinations	Written + Course Work (+ Oral)

A class composed of able students can be offered a complexity of material not suitable for weaker students and the presentation certainly needs to be challenging and demanding; the aims of a course for the less able must be less ambitious and the pace slower. Courses containing large numbers of students of a wide range of ability present problems of both kinds. The preparation of the candidates governs what can be assumed and so what needs to be taught in the course itself with a consequent lessening of time than would otherwise be available. Students wishing to take a course to have background knowledge need different topics and different treatment from those whose specialist study a subject is and who intend to take their work in the field further. The size of the class in relation to the numbers of staff and the facilities can play a big part in forcing the deletion of some topics. For example, supposing in the Computing Science course it was desired to include some laboratory work in circuitry. The number of laboratory places and the possibilities of replication of practicals permitted by timetabling constraints would limit what could be attempted.

The number of teaching staff available clearly restricts the attention that can be given to students in their practical or project work. An even more important feature for all but elementary teaching in Computing Science is the proportion of the topics that should be taught in which the teaching staff have competence. A rapidly developing field like Computing Science poses its own problems and a small number of university teachers in the subject cannot hope to give a balanced and sufficiently complete picture at the usual Special Honours degree level. Other responsibilities weighing heavily upon lecturers in Computing Science is their frequent and deep involvement in other teaching, consulting and cooperation in applied research. However desirable and interesting these other activities are it is only the residue of the teacher's time that can be considered for the contributions they can make to a Computing Science curriculum.

What computer facilities are available will certainly influence how topics are taught and to a lesser extent what is taught. If students must be kept away from the computer at all costs because of competing demands from other users, the sort of experience they can be given suffers some omissions and will need to be repaired in other ways. At this date it perhaps should not be necessary to mention that deficiencies in software need to be taken into account but in many university computing installations it is sadly true. Not merely are the operating systems not always even tolerable but the languages available may not cover those required and the compilers for the ones there are may be less than ideal. If equipment like graph plotters, graphical displays with light pens, Rand tablets, are available, the students can use them. If there is enough data preparation equipment the students can be allowed to do their own preparation and editing. If there are enough key punching staff, punching can be done for them. The laboratory facilities and equipment determine what can be done in practical circuitry and design.

Not the weakest of the constraints, at least in the British degree system, is the need to devise examination procedures which are fair, testing, and which give the opportunity for students of different calibres to demonstrate the class of degree they have earned. The 'degree class' is important for many students, governing as it does, eligibility for many types of official posts and influencing potential employers, perhaps, beyond its proper sphere. The term 'fair' is one which I am sure is well understood but one aspect of fairness is worth stressing, namely that the class of degree awarded to a student should be influenced as little as possible by the impression his or her personal foibles or attractions have had on the teaching staff. It is one of the functions of the external examiner in the British system to do what he can to ensure that the student is neither penalized nor unduly favoured on account of matters outside the subject of his examination.

The objectionable but brilliant student should get his First, just as the idle, pedestrian, but beautiful, female student should get the class her performance merits and not a class above. This seems to me to imply that whatever may be the different forms of assessment used in one examination a substantial part of them should be objective and capable of verification by someone unfamiliar with the individual students. In most university subjects, the written paper and the dissertation have provided material for an outside examiner to study; he might, too, take part in an oral examination. The evidence of this work has been supplemented by practical examination in the laboratory, continuous assessment by means of course work, and laboratory notebooks, in all of which the external examiner must rely more on his internal colleagues. I hope that it is not the observation of a cynic that if students are aware that they will not be examined on a topic they relax their efforts upon it and divert them possibly to those subjects which will be examined. How one can examine certainly reflects what it is reasonable to include in the course.

All these constraints apply in some measure to all subjects in the university curriculum. An example of how Computing Science as a Special Honours first degree subject in Newcastle is constrained is shown on the right hand side of Table 1. Students are normally only permitted to proceed to Special Honours studies if they have given evidence of a fair level of ability. Among them are usually to be found the most able of the students in a university. They have purposely chosen to follow a course treating a subject in considerab depth; in the case of Computing Science they would normally intend to spend at least part of their subsequent career working in the field. Our course is one recently started, producing its first graduates in June 1969 after two years of full time study in the Laboratory. Their type of preparation in their first year in the university means that the more mathematical parts of the subject need not be withheld from them but, intentionally, there is no requirement that they should be able to advance on the engineering or circuitry side. Such then are our students: similar ones, better and worse, are to be found in most British universities - only few (less than a couple of handfuls) of these universities at present offer Special Honours degrees in Computing Science and perhaps even fewer are really capable of offering a satisfactory course. First degrees should be for the general practitioner, giving an exposure to a wide range of the subject and making it possible for the able graduate to extend his own knowledge into whatever specialized field his interests and duties may take him subsequently. The breadth of our own subject is now so great that such a coverage cannot properly be given by just three or four lecturers unless they are exceptional. It has already been pointed out that almost all lecturers in Computing Science departments have commitments outside the teaching of their own specialist students, and they have, of course, a desire and duty to pursue their own researches. Thus I believe that the Special Honours degree is not one that should be offered by a university with a small department of Computing Science. The staff needs to be large and to have a range of interests. The number of lecturing staff in Newcastle is shown in the Table; these are supplemented by some two dozen systems - and research - staff who contribute to the teaching in their special areas.

It is obvious that the computing equipment available in the university will affect what practical work can be offered to the students. This is particularly relevant to the topic of our Seminar - teaching in software. A course on compiler construction will appear to have little relevance to the students unless it is possible to refer to the characteristics of compilers which the student has been able to use. Professor Freeman has described how his course on operating systems is centred upon, although far from restricted to, one system which all the students are supposed to have studied and experienced. The students at S.R.I. on the course given by Mr. Lutz on time sharing systems have as rich a variety of experience offered to them as anywhere in the world and most universities would have facilities which fell far short of those, but some system surely needs to be available so that the specialist students at least can have more than a hearsay knowledge of what is involved.

Finally, this question of examinations has already been raised in discussions in most of the speakers' sessions. It is clear that examining the advanced software teaching is far from easy, especially when it is desired to do more than separate those who should pass from those who should not. Essay type questions give the opportunity to display knowledge and a critical ability but less easily the capability of executing work in the area being tested. For example, Professor Freeman mentioned how he used essays to monitor the reading on historical topics for his course. It is clear from what has been said by all the speakers, and reinforced by our own experience here, that systems software can only be partially examined, at best, by the more usual type of written paper. Parts of both Dr. Ross's topic and Professor Wirth's can provide problems suitable for a three hour paper but, for the most part, another examining method must be found. I was very impressed with the simulated machine project set by Professor Freeman, although in common with others attending I have misgivings that its successful completion is even remotely possible in the time which was available. This would have additional force in attempting to fit something like it into the undergraduate timetable. However, I believe that it is an approach worth following further. In the physical, engineering and biological sciences and in other subjects too, it is regarded as commonplace that there should be technicians, some extremely well qualified, to prepare laboratory demonstrations and experiments for the students to do; their performance in the experiments is assessed by the laboratory demonstrators and by the examiners' evaluation of the laboratory records. So far as I know, in Britain there has been very little, if any, staff available in Computing Science departments to undertake the preparation of software suites for the students to use as a base for their own work. For example, in Professor Freeman's three part project it would be an advantage for the second two groups to have available an approved version of the work preceding theirs. It would, I think, reduce the total time required to something more manageable and also be fairer if the results were to be used in a course assessment. More generally, whenever large and complex systems are being studied a software testbed could play a useful part in the students' instruction and practical work.

The accumulated experience of teaching Computing Science at degree level, and particularly software at this level, is growing but still small and the pace of development of the subject does its best to render all but the latest experience of little value. If the standard of this teaching is to climb it seems vital that both the teaching and examining methods should be made known to those involved elsewhere. This Seminar has, I hope, helped in the first of these; for the second, some exchange of examination papers and projects would be valuable. Some nine or ten years ago we started doing this informally with some of our friends in Britain and we also contributed to what was intended to be a much more comprehensive scheme centred on the other side of the Atlantic. Unfortunatel input far exceeded output which in some circumstances may be good computing practice but in this case was a disadvantage. The journal BIT and some others have published examination questions but these have been mainly confined to numerical analysis, and to Algol programming, both areas which are now among the easier to examine and for which reasonable reservoirs of questions and ideas for questions exist. This is far from a suggestion that these publications should cease, but rather that they should be supplemented by coverage of those topics in which questions are at present scarce. It is time now for another effort to institute a more general circulation of examination papers to lighten the not inconsiderable task of examining and so release additional time for staff in Computing Science departments for their own studies.