Jeff Brian, in your preferred world, what preparation would you like for students entering the University who are planning to study computer science?

Brian Well actually I don't care very much. I would be perfectly happy to have students coming in not knowing anything about computers. That would be OK. We can handle it. I would like to see students with some mathematical sense; not so much knowing any particular mathematical thing but feeling comfortable with the idea of formal notation, and thinking about manipulating symbols in a formal way and both being able to do it and liking doing it too. One thing that I find is that students come into computer science courses thinking that computer science is the study of programming languages or something. Then they are very surprised when they hit theoretical computer science and we start introducing them to mathematical theory about computing, and they say 'what's this got to do with it?'. If they are going to learn to program I'm happiest if they have some exposure to the ideas of functional programming. That is to say, the idea of a program as representing a function. Certainly input/output functions: like for example, starting with some values, cranking them through and getting a result. And I want them to have the ability to combine functions to take the result of one function and use that as the input to another function. I see this in contrast to the more traditional, sequential, first do this, then do this, then do this style of programming. But even that isn't such a big deal.

Jeff Is there anything in students' high school experiences that's bringing them to a certain viewpoint about computing that gives them inappropriate expectations about tertiary computer science?

Brian Well, I don't know if it's their school experiences so much as their kind of street experiences. They have computers at home and the computer probably came with a BASIC interpreter and they've been playing around with it. And then they read in a magazine about other programming languages and maybe they've gone off and learnt Pascal or they've learnt C they've done some combination of things. Or they may have even learnt something I actually like, like Lisp. But this is all about programming and the thing is that I don't want to be understood as thinking that that's bad. I think it's great for kids to program. It's a lot of fun and they should do it. It's just that in addition to that they need to enjoy more formal mathematics.

Jeff You like children to write computer programs. In certain educational circles that's a very unfashionable statement at the moment. There's a conventional wisdom that you don't have to learn to program the computer, that programming is the wrong use of computers in the pre-tertiary school setting. Could you expand a bit more on why you would like children to write programs?

Brian Yeah. The conventional wisdom has gone back and forth several times on this. The first idea was that programming computers was too hard. And then programming computers was the greatest thing in the world and furthermore everybody was going to have to do it. And now the idea is that nobody is going to have to do it. I think all of those points of view are much too narrowly focused on vocational training and trying to second guess what people are going to need in their jobs twenty years from now. One thing is, I don't want to argue that every kid should do anything in particular in high school. I can see some things like that in primary school, you know, everyone should learn to read. I think it's good if everyone has some exposure to beginning programming, back in elementary school. As far as high school is concerned my own feeling is that it's a great time for apprenticeship. I talk about that word a lot. This idea goes really against the grain much more than anything specifically about programming. Kids, teenagers, very often go tremendously hooked on something or other, and I think it almost doesn't matter what it is. Our job is to encourage that interest and try to encourage them to take the interest seriously, whatever it is.

It turns out that not every kid, but a lot of kids, get excited about computer programming. Instead of either saying 'well that's right because every kid should program' or 'it's wrong because nobody should program', what we
should do is take those kids who enjoy programming and give them access to expertise. First of all so that they learn to do it right. Then we should provide an environment in which they have serious work to do. That is, not doing exercises to prove to the teacher that they can do it, but solving real problems. A lot of kids in school get the kind of experience I'm thinking of in sports. They join a team of some kind and their work is taken seriously. They're not just getting a grade for something. They're doing something that people really care about. Some kids get it in the school newspaper. It's very very rare for a kid to have that kind of experience in anything scientific or technical or mathematical.

Jeff In language arts, students refine their work for publication as a standard and proven part of the curriculum. Are you proposing a similar strategy for programming in particular and science in general?

Brian I can think a lot of examples of public performance, that's true and that is part of it but what I'm really on about is the reality of the work. When I was at school they used to have kids who went around and made sure the movie projectors worked when a teacher wanted to show a movie. That wasn't a terribly creative thing to do or anything but it was real work. Now because it wasn't terribly creative it would be foolish for a kid's entire school experience to be about making a movie projector work, but nevertheless that really stood out for a lot of people as something much more real than all the stuff that they were learning. It would be good to have that kind of experience about something technical so that kids who might develop an interest in that direction have the opportunity. And I don't think it matters if they end up being professional computer programmers when they grow up or not. Maybe they'll end up being a physicist or something instead. But it's pretty hard for a kid to have that kind of apprenticeship experience doing physics in school because although you can learn about physics that somebody else did, it's hard to do new physics unless you have cyclotrons and electron microscopes and all that kind of stuff. Whereas it's pretty easy, relatively speaking, to give kids that kind of experience about programming. Although you need computers, mostly what you need is to be at the cutting edge in software, and software is affordable and easy. So I think that it's the reality of the experience of being able to write a program that somebody else uses that really counts. When I taught high school I had a bunch of kids who spent close to all their time in the computer centre, and I would encourage them, even if they didn't do so well in their other classes as a result. I thought it was more important. They could learn the other stuff some other time.

Jeff What have you got to say to the social critics of computer use? Those who on hearing a story of the students spending all of their time in the computer centre see an image of children addicted to the machine.

Brian Yes, there are really two different criticisms like that. One is about the whole idea of concentration. The first thing I would want to know is, suppose instead the kids were passionately interested in art? They went around drawing everything all the time, keeping sketch books, taking art classes. Would the critic have the same objection? Someone would and some wouldn't. OK, if they would; if it's a question of kids shouldn't concentrate - it's kind of funny because when it comes to college students or undergraduates, which is who I teach now, I take the opposite view. I think undergraduates should have a broad liberal education and this is one of the things I argue about with my colleagues. I teach a lot of students who are signed up for engineering degrees and they know nothing about engineering courses. I feel that they are not getting an education. But I think that high school educates teenagers at a time when it just makes a lot of sense to do a specialisation. It fits in with what they're reasonably well educationally and socially at that age. The kind of intensity and idealism that high school kids display fits in well with focusing on something and at the same time they are not ready yet for some of the broader things. Kids take high school subjects even if they are not really interested in them, and even if they get A's in the course, the don't see it as connected with their life in any way. They will be ready for that a little later.

I sort of see both sides. I would agree that people shouldn't get so narrowed down permanently that they don't get a broad education, and if that's what the critics' fear is I agree with it. But that sort of temporary narrowing down at the high school level isn't problematic for me. I tend to push it maybe to an extreme because I'm reacting to the kind of lock step, 'everyone has to learn these six subjects through high school' curriculum that is much more common. OK, now that other kind of objection is specifically about computers. It's the notion that there's something about machinery that's dehumanising. All that I can say is that it can be and it doesn't have to be. Actually, I think, when it looks like that it's probably more the other way around. Somebody who somehow isn't getting along well with human beings finds the possibility of working with computers as a relief, at least they can do something right.

Jeff So it's a symptom rather than a cause?

Brian Yeah, but I've mixed feelings about that too. I think it requires sensitive understanding on the part of the teacher. It's something for teachers to watch out for. I think it can turn out to be a good thing. With some of my high school students I've had parents of kids come to me and say 'the computer centre is the best thing that ever happened to my kid because he used to not have any friends and now he does'. It can be a way in to human contact. If you find people who are the same kind of person you are, you can branch out after that and feel a little more confident with people. One of the things that I used to see that was very amusing were boys talking to each other through the medium of the computer because there were things they wanted to say to each other that they just couldn't say face to face. That was sort of an interesting experience — I don't know whether it's for everybody. Certainly I think that the idea that if you use machinery as a tool you have to be like a machine, that's just not so.

Jeff What is for everybody? Aside from those students who have either aptitude or interest for programming that you obviously want to encourage. Pragmatically schools have to face up to some sort of core curriculum. A vogue at the moment is to describe computer use in schools as information technology. Have you got a view on what is an acceptable core for computer use for all children?

Brian My feeling is that a lot of the fuss about this is just an immediate temporary problem coming from the fact that the 'grown ups', the people who do the curriculum planning, grew up in a world without computers. It's all new to them and they're a little panicked about the situation. Kids grow up using computers and are much more relaxed about the whole thing. Certainly computers are a terrific tool for everybody. I don't understand how anybody ever wrote anything before there were word processors. I've
written a series of books and I certainly could not have done it with typewriters and so on. Every kid ought to have the opportunity starting in the very very earliest grades. By the time they get to high school they should just know. If they don’t know, we should teach them quickly. I don’t like word processing courses, spending a whole semester learning to do word processing. There’s just not that much to it.

**Brian** Much the same thing. I think that they should be available to everybody from early grades. One thing that there is an argument about is whether a kid should have the same software that ‘grown ups’ have or whether we should design special simplified software for kids. I tend to lean to real software for kids. Without making it a big fuss kids should just have these tools available. A ninth-grade class spending two weeks learning how to use a word processor or learning how to use a spreadsheet or whatever is quite enough. I just don’t think it’s a big subject to make a big fuss about.

**Jeff** What’s your advice to any student who does have the interest and aptitude to go on beyond secondary school with computing? What should they focus on?

**Brian** Well, I teach a course at Berkeley that was developed at MIT by Hal Abelson and Jerry Sussman. The course is called ‘Structure and Interpretation of Computer Programs’. It’s taught in Scheme which is a little unusual. It’s a language that comes from artificial intelligence work, but the point isn’t about the language. The course talks about big ideas, about different ways of thinking about organising a computer program. There’s functional programming which I mentioned before, object oriented programming which is a very hot buzz word these days, logic programming and ideas like that. The more typical course that you get even at the university introductory level spends a lot of time on the particular grammatical rules of some programming language, and I think that’s a terrible waste. The kind of programming that you do if you are a kid and you are on your own with a computer really is perfectly good as long as you are writing programs that can sort of fit in your head all at once. When you want to deal with very large complicated systems you start needing help at being able to deal with one piece of it at a time, so you can isolate one piece and work on that separately and still have them all fit together properly. Those structuring ideas are really what we are trying to get across in computer science.