LFCS Education

Course Brochure 1987

Laboratory for Foundations of Computer Science
Department of Computer Science - University of Edinburgh

> *"It is reasonable to hope that the relationship between computation and mathematical logic will be as fruitful in the next century as that between analysis and physics in the last. The development of this relationionship demands a concern both for applications and for mathematical elegance"*
>
> *John McCarthy, 1967*

In the two decades since McCarthy made the statement above the research groups which now form the Laboratory for Foundations of Computer Science (LFCS) have been at the forefront of research in the foundations of Software Engineering. Many current formal methods rely on theories and techniques discovered or invented in Edinburgh. The mathematical basis (and a certain amount of elegance!) has matured in this 20 years. With the foundation of the Laboratory we recognise the need for the interaction of application and theory. The Laboratory's programme is designed to allow software engineers, from both industry and academia to acquaint themselves with methodologies and techniques which will be foundational to software engineering practice during the next twenty years.

The Course Programme

Inevitably with a young subject like Computer Science, and SE in particular, the gap between current research and current industrial practice is a large one, difficult to bridge. It is dangerous to adopt the view that the results of SE research should be encapsulated in software systems or in packaged methodologies. Nor can we afford to wait for the passage of expertise from Academia to Industry through new Computer Science graduates, though this plays a vital role in the long term. Industry has become well aware of these shortcomings, while finding it hard to spare its staff for a year to attend Master Degree courses. Therefore short courses of the kind announced here play an essential part in developing expertise in the immediate future.

The work of LFCS involves both fundamental research and the construction of prototype software tools to mediate the methodologies based upon this research. Present research topics include : (1) the theory of a logical framework in which users may use a machine to help them reason rigorously about their subject, in particular systems and specifications, in whatever logic is most appropriate; (2) the design and semantics of programming and specification languages in which these systems and specifications are to be expressed; (3) program-based methodologies machine-assisted reasoning, notably the techniques of proof-finding and proof-editing; (4) the theory of concurrent systems, both hardware and software, which can be applied for example to the analysis of communications protocols or to hardware verification; (5) software tools for analysis of concurrent systems, with a graphical interface and a rigorous basis in the theory.

The course programme announced here aims to transfer this knowledge in its current state to Industrial practitioners, to equip them for designing and using rigorously based methodologies; as a bi-product it will also provide valuable feedback on the research in progress. Courses of this kind on Functional Programming and on Concurrency have been running for several years with considerable success. The programme is now extended to include Denotational Semantics, Specification, and Machine-assisted proof; in future years it will grow further.

The Laboratory has a world-wide reputation in this field of study, and the courses will be run by its first-class research team consisting of about 8 full-time lecturing staff and 12 postdoctoral research fellows, assisted by some 15 Ph.D students. These people are concerned not only with the research but also with transmitting its results and in having them used.

LFCS Courses 1987

FUNCTIONAL PROGRAMMING IN STANDARD ML

Mon 11 May - Fri 15 May

Presenters: Robin Milner, Robert Harper, David MacQueen, Don Sannella

Standard ML is a functional programming language designed to work well for large programming projects. Its polymorphic types system (first introduced in ML and now adopted for other languages) has been found to yield both convenience and error-freedom in programming. It also provides imperative constructs (exceptions, assignments) which are essential in large systems.

The course is an introduction to the design, formal specification and implementation of software systems using Standard ML. You will learn to use the Standard ML "Core Language" which provides advanced features for constructing small-to medium-scale systems, and the Standard ML Module Facility for assembling program modules into medium-and large-scale systems. You will also learn methods for formally specifying Standard ML programs and the interfaces of Standard ML modules. The topics discussed will be liberally illustrated with examples and considerable time will be devoted to practical sessions.

Fees:	£500	Academic:	£400
Academic Standby:	£250	Student Standby:	£120

ALGEBRAIC SPECIFICATIONS IN THEORY AND PRACTICE

Wed 11 November - Fri 13 November

Presenters: Don Sannella, Andrzej Tarlecki

This course is an introduction to the specification and formal development of software systems using the algebraic specification method. The main emphasis will be on introducing basic concepts and methods for the specification of small-and large-scale systems, while at the same time a glimpse into the theory which underlies these methods will be given. Individual and group exercises will allow students to practice the application of these methods in the construction of software systems.

Fees:	£300	Academic:	£240
Academic Standby:	£150	Student Standby:	£75

DENOTATIONAL SEMANTICS

Wed 10 June - Fri 12 June

Presenters: Stuart Anderson, Colin Stirling

Language design is a pervasive activity in Computer Science. Even in the simplest of application programs we are concerned with the design of both the appearance and structure (syntax) of the input and how that structured input should be interpreted (semantics). Denotational Semantics is the study of such interpretations. Although the techniques of denotational semantics were developed primarily in the study of programming languages the ideas have more gereal utility in the design of software systems.

In this course you will be instructed in the basic language design techniques of denotational semantics. You will learn how to give interpretations to many standard language constructions, for example: sequencing, use of context, jumps and exceptions. Illustrations will be drawn both from programming languages and from design of other systems. A number of small language design exercises will give you an opportunity to practise language definition techniques.

Fees:	£300	Academic:	£240
Academic Standby:	£150	Student Standby:	£75

PRACTICAL INTERACTIVE PROOF-EDITING

Wed 16 September - Fri 18 September

Presenters: Rod Burstall, Tatsuya Hagino, Claire Jones

This course aims to give participants hands-on experience with an interactive proof editor for first order logic with induction, developed at Edinburgh by Brian Ritchie et al. The system runs on Sun-3 work stations and has been designed for ease of use by non-logicians.

There will be plenty of practical work , with one workstation per two students. Participants can take back copies of the system for use in their own companies. There will be a general discussion of interactive theorem proving, including LCF and Nuprl. The course will be of value for people who want to get a feel for the capabilities and limitations of current systems for machine assisted proof of mathematical theorems, notably the sort of theorem which might arise in program verification.

Fees:	£300	Academic:	£240
Academic Standby:	£150	Student Standby:	£75

FOUNDATIONS OF CONCURRENT SYSTEMS

Mon 21 September - Fri 25 September

Presenters: Robin Milner, Kevin Mitchell, David Park, Colin Stirling, David Walker

The course is an introduction to the formal specification, design and verification of concurrent systems using the Calculus of Communicating Systems (CCS). You will learn how to use the language of CCS to model systems of communicating processes, and how, by describing both a system and a specification of its intended behaviour in CCS, it is often possible to determine whether or not the system meets its specification. The techniques for so doing will be illustrated by examples, including the modelling of communications protocols. The criteria to be used in determining whether a system fulfils a specification will be considered. The language PFL (parallel functional language) will be demonstrated in use as a simulation aid for CCS, and the Concurrency Workbench (currently under development in Edinburgh) will be used as an analytical tool. The translation of a parallel imperative language into CCS will be presented, and the course will conclude with a study of modal logics for expressing properties of CCS processes.

Fees:	£500	Academic:	£400
Academic Standby:	£250	Student Standby:	£120

Future Courses

Late 1987 - 1988

Laboratory staff are experienced in many areas of computer science and it is planned to increase the range of courses offered. It is hoped to provide courses in some or all of the following areas during the next two years:

- Advanced Concurency
- Novel Language Implementation
- Modular programming and Abstract Data Types
- Type Theory
- Non Standard Logics
- Domain Theory

The Laboratory

The Research groups which now form LFCS have a long record of effective research in the foundations of Software Engineering, in the broad sense which aims to provide a rigorous basis for analysing specifications and verifying systems. Firmly established methodologies and software tools, originated in Edinburgh include: the specification language CLEAR the ideas from which have promoted a number of further developments such as VDM and Larch; the proof assistant program LCF (Logic of Computable Functions), which pioneered the concept of interactive proof; the Calculus of Communicating Systems CCS which underlies the LOTOS protocol description language; the functional programming language Standard ML, which in mathematical terms is probably the best understood programming language in large scale use today.

Generous funding from SERC and Alvey has allowed LFCS to intensify its research effort; the Laboratory now has some 45 members and is probably the largest group in Europe working on theories for SE. This work aims specifically to underpin the SE discipline of the next two decades (say) with the right basic theory and methodologies. The Laboratory is therefore strengthening its contact with Industry. Significant joint projects with ICL, BP, SSL, STC have been mounted in the past; these are now being augmented by broader interactions.

First, LFCS is introducing a varied programme of courses, announced in this brochure. Second, an Industrial Affiliation Scheme will allow Industrial groups - small or large - to maintain close contact with the research work. Third, the Laboratory will accommodate a few visiting Industrial Scientists for periods of a few months. In this whole endeavour, the guiding principle is that Industry urgently needs to understand the basis of SE, not merely to exercise formal methods by rote, and that a research Laboratory equally urgently needs feedback from Industry to monitor its progress.

LFCS Courses 1987

Registration Form

I wish to reserve a place on the course below and enclose a cheque/purchase order for £.............

Course name...

Name...

Address...

....

....

Telphone...

Signed..........................Date................

Return this form to:

Edith Field
Training Course Co-ordinator
UnivEd Technologies
16 Buccleuch Place
Edinburgh EH8 9LN

Telephone: 031 - 667 - 1011 ext. 6742
Telex: 727442 UNIVED G

LFCS Courses 1987

Registration Form

I wish to reserve a place on the course below and enclose a cheque/purchase order for £.............

Course name...

Name...

Address...

....

....

Telphone...

Signed..........................Date................

Return this form to:

Edith Field
Training Course Co-ordinator
UnivEd Technologies
16 Buccleuch Place
Edinburgh EH8 9LN

Telephone: 031 - 667 - 1011 ext. 6742
Telex: 727442 UNIVED G

Terms of Business

Fees

The cost of the course is £500 (plus VAT @15%). This includes lunch and refreshments as well as the course dinner scheduled to take place on the Thursday evening. Travel and accommodation costs are not included. A discounted price of £400 (plus VAT) is available to all academic staff and students. **Standby** places may also be available for academic participants. These may be booked at any time. Any available places left one week before the start of the course will be allocated in order of receipt of the standby applications. Unsuccessful standby applicants will have their cheque or puchase order returned. Standby prices are: £250 for academic staff and £120 for **non-salaried** research students.

Registration

To secure a place on the course please complete and return the registration form opposite along with either a cheque or purchase order. Photocopy the form as appropriate. 'Full' bookings will be allocated on a first-come-first-served basis. Standby places will be allocated one week before the course starts.

Cancellations

Place are transferrable. Cancellations received within one week of the start of the course will not be refundable. For cancellations received prior to that date a 20% charge will be levied to cover administrative costs.

In the event of circumstances beyond our control, UnivEd Technologies Ltd reserves the right to cancel or re-schedule the course. All fees paid will be refunded.

For assistance with hotel reservations or for any other information please contact:

Edith Field
Training Course Co-ordinator
UnivEd Technologies
16 Buccleuch Place
Edinburgh EH8 9LN

Telephone: 031 - 667 - 1011 ext. 6742
Telex: 727442 UNIVED G
Fax: 031 - 667 - 7938