

Proposed modifications to MSc specialist areas and programming requirement

James Cheney

Abbreviations:

EPL - Elements of Programming Languages (level 10, UG3)

ES - Embedded Systems (level 11, UG4)

ITCS - Introduction to Theoretical Computer Science (level 10, UG3)

APL - Advances in Programming Languages (level 11, UG4)

TSPL - Types and Semantics for Programming Languages (level 11, UG4)

Proposal 1: Additions to MSc specialist areas

The following additions to the MSc specialist areas are suggested:

Cyber Security & Privacy

- Add EPL, APL, TSPL as optional courses

Rationale:

Programming languages ideas are relevant to security in several ways, as already explored in Secure Programming (SP). Though EPL is an UG3 course, it may be useful for students who have not had a similar course in their undergraduate degree (as with DBS, LP in other specialist areas). APL would be more suitable for students already having taken such a course. TSPL covers operational semantics and formalization of language properties in theorem provers, techniques that are now widespread in security.

Computer Systems, Software Engineering & High-Performance Computing

- Add CT, ES, EPL and APL as optional courses

Rationale: EPL and APL have some systems content (programming interpreters; language support for concurrency and databases). ES is a UG4 systems programming course. Though CT and EPL are UG3 courses, one or both may be useful for students who have not had a similar course in their undergraduate degree (as with DBS, LP in other specialist areas). EPL is taught in Scala, an increasingly popular language for distributed and parallel programming. APL would be more suitable for students already having taken such a course.

Theoretical Computer Science

- Add EPL, APL, ITCS as optional courses

Rationale:

ITCS introduces core concepts (complexity, computability). EPL and APL have some theory content. Though EPL and ITCS are UG3 courses, they may be useful for students who have not had a similar course in their undergraduate degree (as with DBS, LP in other specialist areas).

Proposal 2: Allow EPL to be taken to meet the MSc programming requirement

Taught MSc students are required to meet a programming requirement, either by taking one of the following courses:

- IJP (Java)
- LP (Prolog)
- CPSLP (Python)
- Bioinformatics Programming (Python, Java)

or by permission / discussion with their Personal Tutor.

It is proposed to add EPL to the list of courses whose successful completion suffices to meet the programming requirement. (This does not appear to require a change to the DPTs since this requirement is not explicitly enforced there.)

Rationale: EPL includes a practical programming component (currently one lab and two courseworks in Scala). This would be suitable for MSc students that already have some Java or Python experience who might not benefit as much from taking IJP. EPL is similar in difficulty and theory/practical mix to LP. This might be a more attractive choice for students in the *Theoretical Computer Science*, *Cyber Security & Privacy* and *Computer Systems, Software Engineering, and HPC* specialist areas, and might help load-balance IJP.

Proposal 3: New specialist area in Programming Languages & Systems

It is proposed to create a new specialist area (for Informatics and Computer Science MSc programs) called **Programming Languages and Systems**, with suggested courses as follows:

S1:

- Compiling Techniques (Core; optional if already taken)
- Elements of Programming Languages (Core; optional if already taken)
- Automated Reasoning (pending possible change to S1)
- Extreme Computing
- Advances in Programming Languages (Core)
- Logic Programming

S2:

- Formal Verification (in development)
- Compiler Optimisation (Core)
- Parallel Programming Languages and Systems (Core)
- Secure Programming
- Applied Databases
- Introduction to Theoretical Computer Science
- Embedded Systems

with the idea being that students should take 30-40 credits from among the "core courses" and 60-80 overall from this list to qualify for the specialist area.

Rationale:

Tom Ball and Benjamin Zorn recently argued that "industry is ready and waiting for more graduates educated in the principles of programming languages." (<http://dl.acm.org/citation.cfm?doid=2663342>). Recently, UCL has created a dedicated MSc program in "logic, semantics and the verification of programs" (http://www.cs.ucl.ac.uk/degrees/msc_lsvp/), so it seems that they also believe that there is demand for this kind of thing.

The School of Informatics has an international reputation in both compilers, optimization and programming language implementation (multiple faculty in ICISA; recently hosting international conferences such as PLDI 2014 and ParCo 2015) and in functional programming, verification, programming languages theory and foundations (multiple faculty in LFCS and CISA; hosting international conferences such as ICFP 2009, FLOC 2010 and PPDP 2016). So, the School is already in an excellent position to meet demand for MSc students in programming languages. Recent experience has also suggested that there is untapped potential for deeper collaboration between these groups, and establishing a MSc specialist area is one way we could highlight this area to students and increase the number of students having an appropriate background to do PL-related MSc projects (which may in turn feed into the PPar CDT or other PhD programs).

None of the existing specialist areas highlights programming languages per se; while the Computer Systems, Software Engineering and HPC specialist area and the Theoretical Computer Science specialist area together cover the above courses, there is no other single specialist area that covers both PL systems and PL theory.

The resource implications of creating such a specialist area are not clear, and comments are solicited on how to establish (a) whether creating a new specialist area would change the number of students taking various courses and (b) whether these are changes we want.