MPhil in Advanced Computer Science
Software Verification

Leader:	Matthew Parkinson
Timing:	either term
Prerequisites:	Formal Specification and Verification
Structure:	16 Lectures

AIMS

This module aims to provide a detailed overview of the current state of the art in verification of programming languages such as Java and C.

SYLLABUS

1. Control structures: Procedures, Exceptions, and Jumps. (2L)
2. Heap/aliasing — modifying pointer datastructures: Arrays, heap as an array, framing, ownership (Universes and Spec#), and separation logic. (6L)
3. Abstraction and information hiding: Data abstraction, Object/class invariants, and Behavioural subtyping. (2L)
4. Concurrency: Shared variable concurrency, Rely/guarantee, and Concurrent Separation logic. (6L)

OBJECTIVES

On completion of this module students should:

- be able to verify sequential programs involving pointers;
- understand how abstraction enables modular proofs; and
- understand the difficulties of verifying concurrent programs, and be able to use logics to solve different problems.

COURSEWORK

Exercise sheets will be provided.

ASSESSMENT

The course will be assessed by means of a graded term paper.

RECOMMENDED READING

Preparatory reading:

- Chapters 6 and 7 of “The Formal Semantics of Programming Languages” by Glynn Winskel,
- Course notes Part II course on “Specification and Verification I” by Mike Gordon:
  `http://www.cl.cam.ac.uk/Teaching/mjcg/Lectures/SpecVer1/Notes/Notes.pdf`

Combination of course notes + research papers bibliography will be provided.

Last updated: January 2009