This is a technically demanding course in which students learn a new programming and modelling language based on a process algebra FSP and develop models and analyse behaviours of concurrent systems.

In general, the performance of students in the exam was very strong with an overall average reflecting this of approx 68%. The quality of many answer scripts was very high indeed. It was clear that those who attended the course and did sufficient revision did very well, with marks in the 70s and 80s not uncommon. There were however a number of students who attended only a few of the lectures and this is reflected in a group with low (below 50%) to very low (below 35%) marks.

Q1: Modelling and implementation: Here the modelling in FSP was generally well handled but the Java code of a monitor was not, in general, well understood or expressed.

Q2: Proof rules and derivations: Most students understood this well and could construct derivations and labelled transition systems.

Q3: Bisimulation: Most students had clearly learnt the concept well (though sometimes the exact quantification was unclear) and also the algorithm and could apply it. Well done.

Q4: Properties of concurrent systems: Again, students generally understood the idea of properties as processes and how we can automatically verify properties and fix models. The latter was well handled with a variety of solutions suggested.

Q5: Deadlock. Also very well answered: Suggestions for fixing the Dining Philosophers were suitably varied and most were well expressed in FSP. Marks were lost by not referring to the necessary conditions for deadlock.