This is a technical challenging course, introducing a new range of languages (process algebras) and their semantics, their role in modelling concurrent systems, as well as (a) a study of multithreaded programming in Java and its relationship with process algebras, and (b) the analysis of properties of concurrent systems including deadlock and livelock, and safety and liveness issues.

The students performed extremely well, with an average mark near the 2.1/1st class boundary. The standard of the exam was as previous years, but the answer scripts were of higher quality than previously. Whilst the turnout in lectures was moderate, it is clear that a large majority of the students prepared for the exam to an impressive extent, using all the resources that the course made available (past papers, exercise+answer sheets, a course textbook, lecture slides and accompanying notes, and 4 revision lectures. Whether this included the podcasts would be interesting to see.)

Q1 (modelling and implementation in Java) was chosen by few students, and most provided good answers, though some found it difficult to express the relationship between FSP and Java.

Q2 (rules and derivations) - most handled this very well indeed, with almost fully correct answers. Some gave incomplete or incorrect derivations (two applications of parallel rules were required).

Q3 (process equivalence) - again a large number of almost fully correct answers. The processes given were indeed bisimilar, and most applied the algorithm exactly (though some neglected to record all the steps).

Q4 (safety properties) - again many almost complete answers, though some failed to calculate the composite of the process and property correctly.

Q5 (deadlock in concurrent systems) - very good responses, with some impressive FSP solutions. However, some responses failed to present criteria for deadlock correctly, and others to apply these well in the example.