Assessing Practical Skills in a Large Programming Class
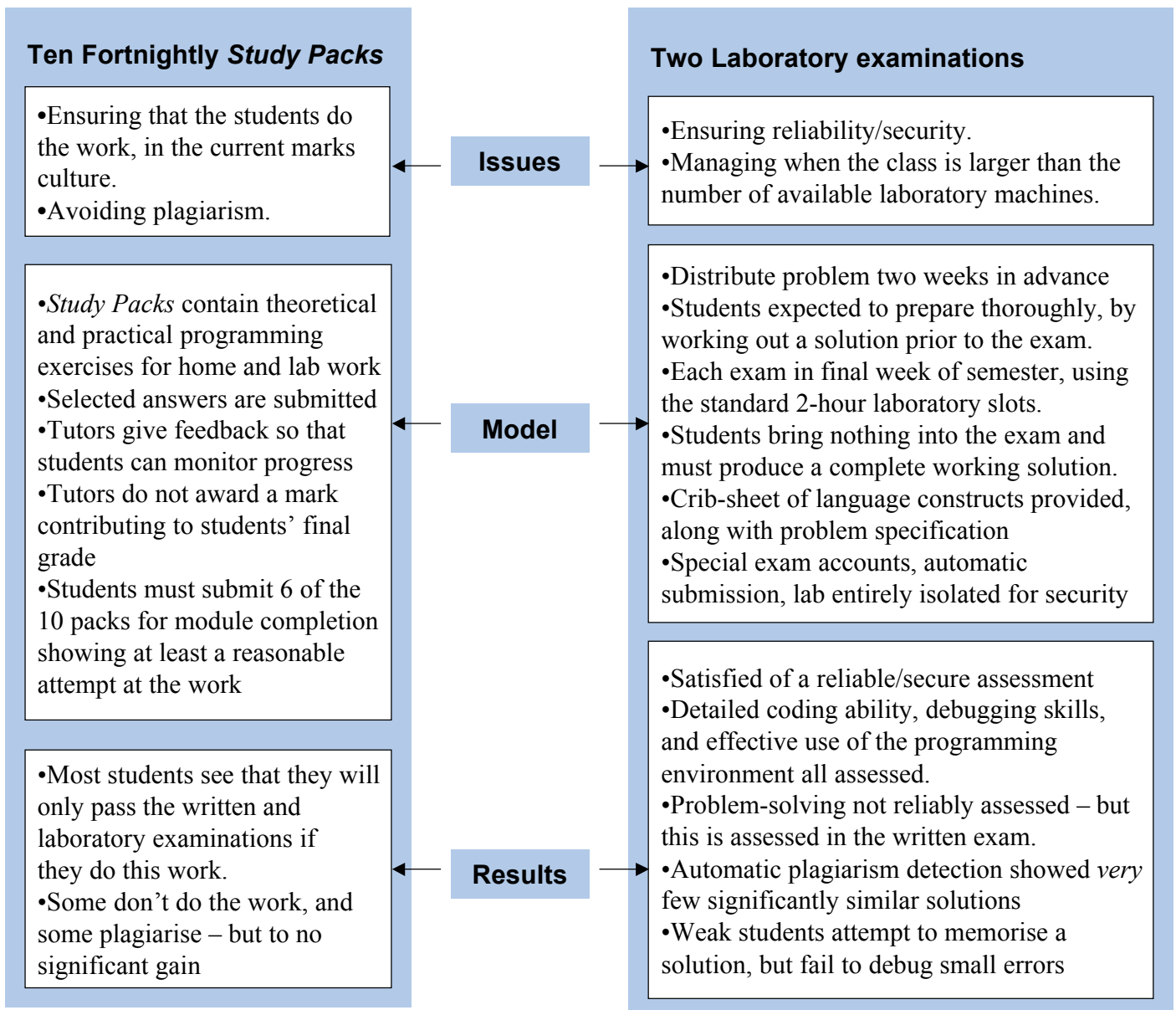
Problem – Continually-assessed exercises are unreliable and disrupt the learning process
Assessment usually consists of a combination of coursework exercises and a written examination.
The examination successfully tests problem solving and high-level coding skills.
The exercises test problem solving, coding and debugging skills, but emphasise **mark-chasing** over learning and understanding, encouraging **stress** and **plagiarism** in the learning phase of the module.

Suggestion – Retain written exam and split coursework exercises into fortnightly exercises and laboratory examinations

Ten Fortnightly *Study Packs*

Two Laboratory examinations

Issues

•Ensuring that the students do the work, in the current marks culture.
•Avoiding plagiarism.

•Ensuring reliability/security.
•Managing when the class is larger than the number of available laboratory machines.

Model

•*Study Packs* contain theoretical and practical programming exercises for home and lab work
•Selected answers are submitted
•Tutors give feedback so that students can monitor progress
•Tutors do not award a mark contributing to students' final grade
•Students must submit 6 of the 10 packs for module completion showing at least a reasonable attempt at the work

•Distribute problem two weeks in advance
•Students expected to prepare thoroughly, by working out a solution prior to the exam.
•Each exam in final week of semester, using the standard 2-hour laboratory slots.
•Students bring nothing into the exam and must produce a complete working solution.
•Crib-sheet of language constructs provided, along with problem specification
•Special exam accounts, automatic submission, lab entirely isolated for security

Results

•Most students see that they will only pass the written and laboratory examinations if they do this work.
•Some don't do the work, and some plagiarise – but to no significant gain

•Satisfied of a reliable/secure assessment
•Detailed coding ability, debugging skills, and effective use of the programming environment all assessed.
•Problem-solving not reliably assessed – but this is assessed in the written exam.
•Automatic plagiarism detection showed *very* few significantly similar solutions
•Weak students attempt to memorise a solution, but fail to debug small errors

Conclusions
- Students who attempt memorisation only score highly if they have effective coding and debugging skills
- The laboratory set-up is expensive – we are developing cheaper methods of ensuring the required security
- Separation of learning and assessment viewed as positive by most students and staff
- We are developing the skill of setting problems of the right size and complexity

Contact: Quintin Cutts quintin@dcs.gla.ac.uk   Project participants: Rob Irving, Paul Philbrow, Mark Meenan