Report on COMP6700, Semester 1 2015

Course Title. Introductory Programming

Convenor and Lecturer. Dr Alexei Khorev

Number of Students. 42

Suitability of Prerequisites. Given it is an entry level course for our masters programs and the graduate diploma of computing there is no prerequisites. There is a shift in the cohort of students with more students having less computing/programming background, this course should also shift to better cater from these “conversion masters” type students. As students who do have a stronger programming background can always be exempt from this course.

Composition of the Student Cohort. 67% of the students taking the course are Master of Computing students, 14% are Graduate Diploma of Computing, 7% Master of Information Technology, and 12% other. (based on info from FAIS).

Relevance of the Course for Students. This course is a key introductory programming course our masters students will generally undertake. The course provides an important foundation. Masters students would normally do this in their first semester. As such this is both a timely and relevant course.

Suitability of Learning Offerings and Assessment. Many of the students appreciated Alexi and his deep knowledge of the topic. Overall the mode of deliver for this course was very standard and there was no significant issues with this. There was some attempt at providing video material which had more limited success, however, this is more of an issue of fine tuning.

Feedback and Availability of Resources. Students were provided general feedback on assignments along with an email regarding their result. For the second assignment this was after the exam, however, this was due to the timing of the scheduling of the exam. Alexi knows the material for this course very well, and this was positively reflected in the student’s short answer feedback. Also the course material and resources provided to the students are complete and have been iteratively fine-tuned over the years.
Percentage of Student Feedback. 50% of the students provided feedback.

**Recommendations**  Given that the proportion of students in the class who have limited programming background has grown over the years this course’s focus should also be shifted to better support this new cohort of students. Alexi’s report mentions that he sees that the learning outcomes of this course should be more than that of COMP1110/6710, however, generally student by the end of COMP1110 will have had an extra semester of programming in COMP1100. It seems more appropriate that we should aim to have the exit abilities of COMP6700 more in line with that of COMP1110/6710. We make the following recommendations.

1. COMP6700 attempts to cover too much material and should focus more on the basics of the learning outcomes. Although Java has grown over the years in the language features it provides, there is no imperative to attempt to teach all of Java. Rather the focus should be much more on the basics of procedural and OO programming with a deliberate attempt to restrict/limit the aspects of Java presented. More time could be spent in lectures providing practical examples shifting the course to make more use of a problem based teaching approaches. The following could be removed from the course: varargs, reflections, details about the JVM, self-balancing trees, b-trees, observer design pattern, topics in block six (including functional program). This would free up time to reinforce the fundamentals of this course.

2. In the 2015 exam some of the theory questions would have been better suited to an advanced course on the design of object oriented programming languages, rather, than an introductory programming course. Questions should be more strategically aligned with learning outcomes and reviewed by a second examiner. Also the practical questions were uniformly difficult. As such it would be better to have some simpler questions that evaluate minimum expected learning outcomes (along with some more difficult questions to evaluate meritorious performance). A larger proportion of the exam could have been aligned better with the basic learning outcomes of the course.

3. COMP6700 should, as much as possible, make use of the teaching material of COMP6710. This includes: schedule, lecture notes, labs, assignments, and exams. Such that COMP6700 is basically a first semester instance of COMP6710. This would facilitate the sharing of resources, ideally making less work for both coordinators. It would
also enable COMP6700 students to access material made available from COMP6710. At some point COMP6700 and COMP6710 should be merged into a single course which is offered in both semester the first semester course just for masters students and the second semester instance combined with our undergrad cohort (it is critical that students entering the masters program have such a course they can enroll into in their first semester). Note in terms of the programs-and-courses description the learning outcomes of COMP6710 does not include GUI aspects, whereas, COMP6700 do. However, the current implementation of COMP6710 does include a GUI component which would clearly satisfy the COMP6700 requirements.