SE 3I03
Group Project
Lukasz Nasiadka (0250285)
Dipali Kapadia (0254777)
November 26, 2004

Adding a Cup of Java to Second Year

Problem

Students in the Software Engineering Program at McMaster University have limited programming skills. Currently students are taught functional (Haskell) and imperative programming (C), but are missing a key component, object-oriented. Software engineers are expected to be knowledgeable in all types of programming languages. The fact that they are not decreases their chances of finding successful employment after graduation. Completion of the software program should yield proficient design skills combined with excellent programming ability.

Objective

We want to provide students with an early exposure to the Java programming language. As a result they would be able to grasp object-oriented concepts and gain skills that are assumed in core courses and the workplace. By adding a Java course into the curriculum, students' programming skills would greatly increase as well as their knowledge of a different language.

Approach

Currently, the only exposure students are getting to Java is about three weeks worth at the end of 2S03. This little amount of time is not enough for the regular student to gain a solid understanding of the language and how it really works. The majority of the students would either memorize the basic syntax for the exam or just not care to learn it at all. It seems that these three weeks could be put to better use in perfecting the students' C programming skills.

In adding this course we are not trying to make Java more important than anything else in the software engineering program. We realize that knowing Java will not sustain a students' career forever. The reason we choose Java is because it is an object oriented language that is dominant right now. Should it lose its popularity, it can be replaced with another object-oriented language.

The structure of this course would be very similar to 2S03, in that there would be weekly assignments for practice, but the students would not be required to hand them in. This style encourages students to practice programming without being constrained to a timeline. Towards the end of the semester, a small group project that requires the application of the object-oriented programming skills gained throughout the semester would be handed in. Lastly, two midterm tests would be used to judge the students' knowledge and progress in the course.

A course on Java could be easily integrated into the second year curriculum by rearranging the current courses. As it stands now, there are six courses per semester in second year. One of the six courses in the first semester is a science elective which would be removed. The removal of this elective would not be an issue for accreditation because it is not mandatory. The elective has already been removed from the September 2005 curriculum and replaced with a math course. We propose 2C03 be moved from second to first semester taking the place of the science elective. This would leave five courses in second semester allowing for the addition of a course on Java.

The addition of this new course would require the department to:

- Select from existing staff members, or hire, a professor to teach the course
- Invest time and manpower in setting up and organizing the course
- Hire a teaching assistant to help with the course (i.e. to do marking and conduct tutorials)
- Make rooms available for various tutorials over the week

The timetable for the completion of the project is set for September 2005. After the first class completes the course, it would be re-evaluated to measure its benefits.

Impact

If this proposal were to be implemented, the major impact would be on the students. First, they would be more equipped to work in industry. This is because individuals will always be needed to support all the Java code being generated in the present. Second, since the structure of Java is largely modular, the transition from design to code would be easier. Having this background would help them in future design courses such as 2AA4 and 3A04, where this knowledge is already assumed.

The university would also feel the impact, as it would have to provide funding to organize and implement the course. However, the benefits would far outweigh the costs, especially in the long run, since more knowledgeable graduates would be produced.

Another factor to be considered is the workload on both the students and professors. The implementation of this course would not considerably increase the level of difficulty of second year on the students since we are simply replacing one course with another. The professors would not face a substantial increase in workload either. All that is required on their part is to create the outline for the course and choose somebody to teach it.

Cost

The monetary resources required to implement this proposal are quite reasonable. Assuming one of the current faculty members could teach the course, the only costs would be to hire TAs and set up the course. Therefore, students' tuition would not increase.