Creating A Separate Introductory Computer Science Course For Engineers: An Experience In Moving From Java To Python

Jason Morrison¹ and Terry Andres²
¹Department of Biosystems Engineering, University of Manitoba, Winnipeg, MB
²Department of Computer Science, University of Manitoba, Winnipeg, MB
Corresponding Author E-mail Address: Jason_Morrison@umanitoba.ca

Abstract – The University of Manitoba (U. of M.) 2011-12 curriculum has a new introductory programming/computer science course specifically for current or future engineering students. This course focuses on teaching fundamentals of programming and computer science through mathematical computation using Python. The need for this course came from a grassroots movement by engineering professors to evaluate the previous course and provide direction and/or assistance to the department of computer science. The committee felt it was time to abandon Java as the introductory language for engineers. This talk discusses the resulting course; the challenges in selecting the version of Python to use; and the difficulties and rewards of the change to a new and previously untaught language at the U. of M.

Keywords: Introductory Programming; Course Delivery

1. INCEPTION

In the spring of 2009 the Faculty of Engineering’s curriculum committee met to discuss the perception that the general first year computer science course was not meeting engineering students needs. The committee reviewed the 2004 curriculum guidelines for computer engineering students [1] only after a lengthy discussion on the perceived programming abilities of students; difficulties of Java as a first and/or only language; and the needed ability for engineers to understand and program numerical computations. The result was to recommend that a first year course in computer science be targeted for engineering students; using a scripting language that is more interactive than Java and that it “present computation as a tool for solving mathematical engineering problems”. The more specific course recommendations included: 1) a focus on procedural programming; 2) that it be preparatory for numerical methods courses; and 3) that a language like Python be chosen and Java not be taught. These thoughts were reviewed and discussed by Computer Science and a new course was approved (COMP 1012—Computer Programming for Scientists and Engineers).

2. IMPLEMENTATION & DISCOVERIES

The course shift from Java to Python was not a paradigm shift as both courses were taught from a procedural programming paradigm. However the brevity of Python code for simple programs is attractive as an introductory language when compared with Java’s longer equivalent programs. Thus the courses are alike in pedagogical objectives with the only change from “entering, compiling, running, and debugging Java programs” to “enter, run, and debug Python programs”.

There is a syntactic difference between Python 2.7 and 3.0 that makes the new version not backwards compatible. The implementation required the selection of a consistent textbook and Python distribution, including mathematical libraries. This was easier to do in Python 2.7, as Enthought provided a free distribution that could be easily installed on both Macs and PCs.

The focus on mathematics of computation has been taken up with vigor utilizing a numerical computation oriented textbook [2]. Assignments have also focused on mathematics with an example being the exploration of different methods of computing roots of quadratic equations including the complex roots. Furthermore the use of Enthought and exposure to its libraries is an excellent match to prepare students for numerical methods courses.

Finally, two lessons can be drawn from the first year: an abundance of interested students can create a demand that results in insufficient lab demonstrators or markers trained in Python; and the additional focus on mathematics and numerical computation has to be undertaken appropriately to ensure marking is for the desired skill set and skill level.

References