INTRODUCTION TO ALGORITHMS AND PROBLEM SOLVING

Michael L. Nelson and David Rice[1]

Abstract *- Language and paradigm choice for teaching computer programming is an on-going debate. We have been using Visual Basic for the introductory course and C++ for the advanced course. Unfortunately, many of our students struggle in the introductory course. We are currently re-engineering our degree. While our introductory course should ideally be language-independent, we feel that students will gain a better understanding if their algorithms are executable. Therefore, we have decided to use the procedural paradigm, with C++ as the implementation language for this course.*

INTRODUCTION

What language and what approach to use in an introductory computer programming course has long been the source of an on-going debate [1, 2]. International College (IC) has been offering a fairly conventional Computer Information Systems (CIS) degree, using Visual Basic for the introductory course and C++ for the advanced course. However, we find that many of our students struggle with Visual Basic in the introductory course, but then go on to do quite well with C++ in the advanced course.

We get three main comments from students in our current introductory course using Visual Basic. First, they find that form development is relatively easy to do. Second, that they understand the basic programming constructs. Third, many students have a great deal of difficulty converting their algorithms into Visual Basic code.

We believe that there is a direct connection between the first and third comments. It is very easy to create forms, even quite complicated ones, using Visual Basic. However, it is then the form that drives the overall structure of the program (hence the concept that Visual Basic is 'event driven'). While general procedures can be written, most of the procedures are a direct result of an event concerning one of the controls on the form. As a result, even a relatively simple algorithm can be 'lost' as it is divided up between the various control procedures.

CHANGING TO C++

We have recently re-engineered our entire curriculum, creating four different specialties within a newly created Computer Information Technology (CIT) degree: Database / Software Development; Web Applications, Network Hardware, and Network Software and Administration. All students will take the Introduction to Problem Solving course, for which we will no longer use Visual Basic. Instead, the course will stress the four basic steps to problem solving (analyze, design, code, and test), with an emphasis on algorithm development using C++.

Two things should be noted about this introductory course. First, we would like to teach the course in a language-independent manner, using algorithms only. However, we feel that it is very important for students to be able to execute their algorithms on a computer so that they receive feedback as to what works and what does not work. Secondly, we hesitate to say that we are using 'C++.' We will actually be using mostly conventional C code, with C++-style I/O (that is, object-oriented programming will not be covered in the introductory course; it will be the subject of an advanced course).

The introductory course will begin in a language-independent manner [3], then go on to how each construct is implemented in C++ [4]. The Visual Basic course will then be able to concentrate on business and database applications after a quick review of the basic constructs.

Students in the Database/Software Development track will then go on to take a semester long course in Visual Basic, which will allow the inclusion of advanced applications since they should already know the basics. They also will take the semester long course on object-oriented development in C++, and an advanced database development course to round out their specialty.

CONCLUSIONS

We believe that this approach, using C++ in the introductory course followed by Visual Basic as a follow-on course, will allow our students to be more successful in learning how to create algorithms to solve real-world problems.

REFERENCES

[1]. Nelson, M.L., Meng, X., Brazier, P., "CS I: Language & Paradigm Issues," *Proceedings of the AoM/IAoM 17th Annual Conference*, August 1999, pp. 353-357, San Diego, CA.

[2]. Nelson, M.L., "Teaching C++ / Teaching Object-Oriented Programming," *Texas Computer Education Association Conference*, February 1998, Austin, TX.

[3]. Robertson, L. A., *Simple Program Design: A Step-by-Step Approach*, Course Technology, Cambridge, MA, 2000.

[4]. Zak, D., *An Introduction to Programming with C++*, Course Technology, Cambridge, MA, 1998.*mplish similar goals.*

[1] School of Advancing Computer Information Technology, International College, 8695 College Parkway, Ft. Myers, FL 33919