Considerations for Selecting a Programming Language to Teach Perspective Teachers

Azad Ali, Indiana University of Pennsylvania azad.ali@iup.edu
Frederick G. Kohun, Robert Morris University, kohun@rmu.edu

ABSTRACT
The purpose of this paper is to identify the factors that influence the decision to select a programming language to teach students enrolled at a graduate level course in education. The paper illustrates the experience of two degree programs: First, the master degree of business education (M.Ed.) at Eberly College of Business and Information Technology – Indiana University of Pennsylvania (IUP), and second, the master of education (M.S. Business Education) at Robert Morris University (RMU).

Introduction

Academic technology programs often struggle with the selection of a programming language for entry level programming courses. An entry level programming course can serve as a launching pad for other more advanced programming courses. It sets the stage to learn more advanced programming topics. At the same time, a beginning programming course can also be taken by students from other non-technology majors. While the non-major students may not have the same interest in preparing to learn advanced programming topics, it must be taken into consideration that many students may have dissimilar backgrounds with respect to students enrolled in technology programs.

This paper illustrates the experience of two programs in selecting a programming language to teach master degree students in a first programming course. The M.Ed. program at IUP and the M.S. Business Education program at RMU both teach programming topic/course for their students enrolled in their respective education master degree programs. The paper begins by explaining the factors that make learning to program a difficult task. It then introduces a programming language named Alice and shows how this programming language addresses identified difficulty factors. It then introduces other factors that usually influence the decision to selecting the programming language. The paper then elaborates on the experience of the two programs at IUP and RMU and how they addressed these factors when selecting a programming language for their students enrolled in their graduate programs. A summary of the paper is presented at the end.

Learning to Program – Difficulty Reasons

Learning to program is considered to be a difficult task to many students. It is estimated that at least 25% to 80% of students drop their first computer course due to the difficulty in learning to program (Carter & Jenkins, 2002). This difficulty is experienced by students enrolled in technology programs other students from other majors who take programming courses as part of their requirements to complete their degrees (Baldwin and Kuljis, 2001).

Other studies have provided a comprehensive analysis of the factors that contribute to the difficulty of learning to program. Dann, Cooper and Pausch (2006) listed four factors that contribute to the difficulty associated with learning and programming: Fragile mechanics of program creation, particularly syntax; the inability to see the result of computation as the program
runs, the lack of motivation for programming and the difficulty of understanding compound logic and learning design techniques.

In a study conducted to suggest steps to simplify learning to program, Kelleher and Pausch (2005) compared a number of programming languages that are commonly used in beginner programming courses. The same study wrote the following about the difficulty of learning to program:

“Learning to program can be very difficult for beginners of all ages. In addition to the challenges of learning to form structured solutions to problems and understanding how programs are executed, beginning programmers also have to learn a rigid syntax and rigid commands that may have seemingly and arbitrary or perhaps confusing names. Tackling all of these challenges simultaneously can be overwhelming and often discouraging for beginning programmers” (p. 83).

The remainder of this section explains in more detail some of the reasons that contribute to the difficulty in learning to program.

**Fragile Mechanics and Syntax**

The mechanics of developing a program are not standardized. It is called fragile because there is no direct and pre-defined way to develop a program. Although program creation usually starts by learning the syntax, some argue that tackling the syntax is not the best way to learn to program. Learning syntax is foreign to some students and, as a result, they may spend a lot of time learning the programming language syntax without context.

**Inability to See the result of Computation as the Program Runs**

In order to see the result of execution from any program, the program more often needs to be error free, to execute, and then, display the output. However, making any program run is not always a simple process. After working through the syntax and making the program start to run, the programmer then faces additional errors that are called “execution errors”. These errors may result from missing variable name or using different field types (numeric or otherwise) for wrong purposes. Such as using non-numeric fields for computations or dividing by zero. After working through all these steps, the programmer still may not be able to see the final program output.

**Lack of Motivation to Learn Programming**

The programming profession is seen by many people as a boring job (Rosamita, 2007). This point of view stems from the notion that programmers sit in front of a computer spending long hours trying to produce output or correct errors that seem to be of minimal importance. Thus, motivation is minimal to take different programming courses that have the potential to become qualified to take on what it seems to be a boring job.

**Alice Programming Language**

Alice is a programming language that was introduced by Carnegie Mellon University and it seems that it has provided the answers to the questions that were raised about the difficulty of programming languages. Alice provides a visual interface that makes it easier to follow, and it cuts down on the syntax and coding.

Alice has increased in popularity for use in first year programming courses at both colleges and high schools. The increasing popularity of Alice as a first programming language is due to the many advantages that it provides over traditional or general purpose programming languages. Adams (2008) noted the advantages to using Alice in introductory programming courses: The
allure of 3D graphics, The Alice IDE. Alice includes a drag-and—drop integrated development environment (IDE) that eliminates syntax errors and that Alice is and Object Oriented Programming language.

When developing a program in Alice, users do not have to type the program. Instead, users pull down objects and align them according with specified commands that are already drawn for the user. As the user pulls a particular object, another dropdown menu appears that gives the user options to choose from. The key here is that there is no room to make syntax errors when using Alice. Instead, efforts can be directed to understand the mechanism and the concepts of the program (Porter & Calder, 2004).

Alice uses the structure of object oriented programming. In Alice, classes are selected to select from and then objects from the classes are pulled to a “world”. As each object is drawn on the world, a visible list of properties and methods can be observed and pulled so to use them in the program.

**Seeing program result**
Programming in Alice enables individuals to see the programming code right after the program ends. As the objects are pulled off the visual library, the programmer is able to see results immediately. There are no syntax errors in Alice, thus programmers do not have to stumble through lines of syntax errors. As a result, programmers using Alice are able to see program result as soon as they finish or complete the program.

**The issue of motivation**
Alice uses visual output. All objects within Alice are three dimensional visual objects. The output that is usually generated from a typical Alice program, as a result, is more visually appealing. The objects represent popular metaphors which tell stories, draw shapes, and have moving components. These movements on the screen provide an interesting application to the programmer. Using Alice in a first programming course may be able to solve many of the problems associated with teaching programming for high school students for these reasons:

- Alice dispels the common notion that pervade high schools: learning to program is considered to be asocial (Rosamita, 2007). Such a notion discourages most students in general (high school students in particular) from getting enrolled in computer programs.
- Using Alice, programming teachers can incorporate toys, games and other activities (Anewalt, 2008) that high school students want to have in their class more than other students.
- The Alice application is more likely to have characteristics that interest high school students more often than others, it uses characters that most high school students relate to. (Guibert, 2002).

**Other Institutional Consideration**
Selecting a programming language that simplifies the learning of students could help attracting more students into these courses. However, this is not the only factor that decides on which programming language to select in beginner programming courses. Instead, different other factors influence the selection of such programming language in entry level courses. These following reasons may influence the selection of a programming language for this kind of a course:

- **General Purpose versus Beginner Programming Course.**
- **Major only course versus service course.**
- **Entire Course versus Selected Topics.**
- **Standalone Versus Prerequisites**

## The Two Graduate Programs

This section explains about the two courses that teach the programming course/topic for their master degree students in education at Indiana University of Pennsylvania (IUP) and Robert Morris University (RMU).

### The M.Ed. Program at IUP:

The Technology Support and Training (TST) department at Eberly College of Business in Indiana University of Pennsylvania (IUP) offers a master degree program in business education (M.Ed.). One particular course that is included in this master degree program is a capstone course called “BTST680 Technical Update”. The course teaches the latest in technology and includes four categories or sub-coverage areas: Programming, Database, Digital Media and Networking. The following describes the selection of a programming language for this course and the methods in which it is being taught.

Alice programming language was selected for this course to teach the programming topic. The main reason for selecting Alice is that the students in this course are perspective teachers. Therefore it will be useful to teach them this language as they may need it for their professional lives. The students in this course are not looking for a programming job in the industry; hence it will not help them to teach a general programming language such as Java or C++. Instead, they can focus on learning the concepts of programming by using the tools available in Alice.

The faculty member teaching this course has been using Alice for the past two years. The feedback from selecting Alice in this course has been positive and enrollment has increased in this course since introducing Alice it. The learning curve in the course has also increased. The students master the programming concepts quicker as compared to previous semesters. Students are required to complete and present a final comprehensive project with Alice. All presentations have been successful while the students showed good understanding of the programming terminology such as objects, properties, methods, encapsulation, and inheritance.

### The M.S. in Business Education Program at RMU

Robert Morris University (RMU) offers a master degree program in Business Education. This program requires their students to take one programming language. The programming course is taught by the college of Information Systems and Communication. The title of the course is Visual Basic Programming and the course number is INFS6120.

The Visual Basic programming course is required by students who are enrolled in the technology programs at RMU. Though this course is not a formal prerequisite to other courses in the program, but there is a common understanding that this course is the first programming course for students enrolled in the M.S. Internet Information Systems program, M.S. Information Systems Management program, and other technology programs.

Reactions from students taking this course have been mixed. Technology majors most often have positive view of Visual Basic because it is perceived as a simpler language to learn that others like Java and C++. The same students also view that Visual Basic has wider range practical applications especially in terms of the various connections it provided with the different databases and web browsers. However, business education students do not see it the same way.
To simplify the comparison between the two programs, table 1 below shows the information related to the selection of a programming language at both Institutions (IUP and RMU) as pertains to the factors listed in this section.

<table>
<thead>
<tr>
<th>Program Name</th>
<th>IUP</th>
<th>RMU</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Purpose Vs. Beginner programming</td>
<td>Beginner programming language (Alice)</td>
<td>General purpose programming (Visual Basic)</td>
</tr>
<tr>
<td>Major only course versus service course</td>
<td>Major in Business Education only course</td>
<td>Required by technology majors, business education majors as well as other majors</td>
</tr>
<tr>
<td>Entire Course Versus Selected Topics</td>
<td>One programming topic among four selected topics</td>
<td>Entire course teaches programming</td>
</tr>
<tr>
<td>Standalone Versus Prerequisites</td>
<td>Stand alone. Does not serve as a prerequisite to other courses</td>
<td>Though not formally a prerequisite, there is understanding that this a first course for other courses</td>
</tr>
</tbody>
</table>

**SUMMARY**

This paper discussed the issues and challenges that face the decision to select a programming language to teach for students enrolled in a master degree program in education. It began by explaining the factors that make learning programming languages a difficult task for students. It then introduced a programming language that is intended to provide solutions for the points of difficulty that accompany the learning of how to program. The paper then elaborated on the experience of two graduate degrees in education that teach programming in their respective programs. The Master of Business Education at Eberly College of Business at Indiana University of Pennsylvania and the Master of Science degree in Education at Robert Morris University.

**REFERENCES**


