Work in Progress: Programming Knowledge – Does it Affect Success in the Course "Introduction to Computer Science Using Java"

Jeffrey S. Rosenschein [1], Tamar Vilner [2], Ela Zur [3]

Abstract -

At the Hebrew University of Jerusalem, the course "Introduction to Computer Science" (CS1) is taught using the programming language Java. In Israeli secondary schools, Computer Science is not a required subject, but those who do study it learn a procedural programming language at an advanced level. Some students who study Computer Science at the university level have thus previously been exposed to the field

Our research examined whether prior knowledge of programming languages contributed to the success of students in their first university programming course. The research included an examination of the correlation between prior knowledge and their success in the course. We queried students both about their familiarity with various programming languages, as well as about a variety of programming concepts. A partial analysis of the results shows that there is a positive correlation between having previously learned procedural languages (such as Pascal and C) and success in the course

Index Terms - CS1, OOP Programming, Procedural Programming,

INTRODUCTION

The Hebrew University of Jerusalem is one of Israel's leading universities, and its School of Engineering and Computer Science produces research of the highest international standards. Undergraduate admission requirements to the School are particularly strict, and students must have demonstrated excellence in their high school studies, matriculation exams, and aptitude tests, to be accepted.

The first course in the Computer Science undergraduate program, "Introduction to Computer Science" (CS1), is based on an object-oriented programming (OOP) approach using the Java programming language. Topics covered in the course include a) flow control: conditional statements and loops; b) variables and simple data structures, such as arrays, linked lists, stacks, queues, and binary trees; c) explanations about the computer itself: hardware and software concepts; d) the foundations of OOP: objects, classes, methods; e) more advanced OOP topics: inheritance, static methods and variables, method overloading, packages, polymorphism, and interfaces; f) recursion; and g) basic sorting and search algorithms.

Many students entering the Computer Science program have prior programming experience. Some study programming in high school, some in various courses, and others study on their own. There exists research claiming that object-oriented programming is particularly difficult for beginning students, and that it is preferable to teach this approach only after students have been exposed to procedural programming.

THE STUDY

We are in the midst of a study aimed at examining whether prior programming knowledge affects success in the course "Introduction to Computer Science" at the Hebrew University. Prior knowledge will take into account various programming languages as well as knowledge of computer programming terms (conditional statements, loops, complexity, objects, inheritance, and more). We will examine the effect of prior knowledge on success in the various course assignments and in the final exam.

The research questions are:

- To what extent does prior knowledge of procedural programming affect students' achievements in CS1?
- To what extent does prior knowledge of object-oriented programming affect students' achievements in CS1?

I. Research Tools

During the Fall semester of 2003, about 100 students took the CS1 course. At the beginning of the semester, a questionnaire was administered to determine students' prior knowledge of computer languages, and their knowledge regarding computer programming concepts.

One question related to prior programming knowledge, indexed to various programming languages (Basic, Visual Basic, Pascal, PROLOG Assembler, C, C++, Java). For each of these languages, the student was to state whether they had previously learned it (either independently or in a course), and also to rate their own knowledge of the language on a scale of 1 to 5 (where 1 signified "Do not know the language at all",

[1] Jeffrey S. Rosenschein, The Hebrew University, Givat Ram, Jerusalem, Israel 91904, jeff@cs.huji.ac.il
[2] Tamar Vilner, The Open University of Israel, 16 Klausner Street, Tel Aviv, Israel 61392, tami@cs.openu.ac.il
[3] Ela Zur, The Open University of Israel, 16 Klausner Street, Tel Aviv, Israel 61392, ela@cs.openu.ac.il

and 5 signified "Know the language well, and can program in it without help").
The second question related to programming concepts, such as variables, conditional statements, loops, functions, recursion, pointers, object, class, constructor, inheritance, polymorphism, etc. For each of these concepts the student was asked to describe their level of familiarity with, and knowledge of, the concept on a scale of 1 to 5, where 1 signified "Do not know, do not understand", and 5 signified "Understand well and could explain to a friend".
It is important to note that the students' answers subjectively described their levels of knowledge, and we did not verify this in any other (objective) way.

II. Partial Findings

Our research is ongoing, and we will therefore present only some of our results. From the questionnaires it became clear that a large number of students had previously learned a programming language. Particularly prominent was the number of students who had learned Pascal (68\%). Other languages were not as widely studied; PROLOG (for example) had been studied by only 5\% of students. Figure 1 shows the percentage of students who studied each language.

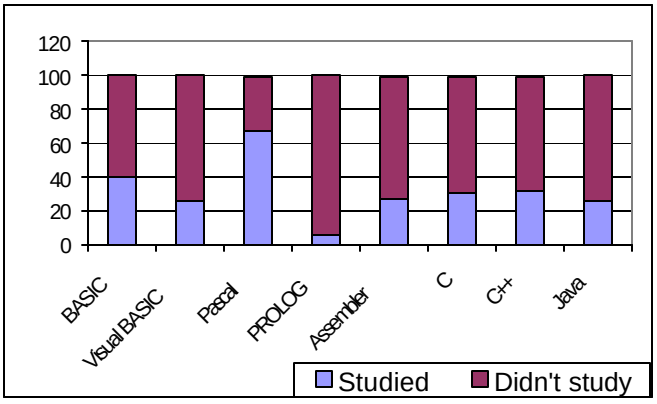

FIGURE 1
Percentage of Students Studying Computer Languages

Table 1 shows the knowledge students claimed to have of each language, according to the 1 to 5 scale. Here, too, one can see that students' professed knowledge of Pascal is highest among all the programming languages.

TABLE 1
KNOWLEDGE OF PROGRAMMING LANGUAGES

Language	Mean	Std. Deviation
Basic	2.58	1.227
Visual Basic	2.37	1.222
Pascal	**3.42**	1.366
PROLOG	1.49	0.753
Assembler	2.20	1.093
C	2.66	1.282
C++	2.74	1.277
Java	2.59	1.180

At the end of the semester, students were tested and received grades on a scale of 0 to 100. We examined the correlation between prior knowledge of each programming language, and success in the course.

TABLE 2
CORRELATION BETWEEN KNOWLEDGE OF PROGRAMMING LANGUAGES AND SUCCESS IN CS1

Language	Pearson Correlation	Significance
Pascal	0.377	< 0.001
Assembler	0.234	0.026
C	0.216	0.041

We found a positive correlation between various languages and success, as described in Table 2. No significant correlation was found for languages not appearing in Table 2.
For lack of space, we will not present here the frequency of students' level of knowledge regarding each of the programming concepts that we checked. In any case, to our surprise, we did not find a correlation between prior knowledge of any of the concepts and success in the course. However, there may have been "noise" in the data set (at least with regard to object-oriented concepts), since students responded to the questionnaire after one lecture on the basics of OOP, and may have evaluated their knowledge artificially high after brief exposure to these concepts.

PARTIAL CONCLUSIONS

Students in the Computer Science program at The Hebrew University of Jerusalem are accepted after a careful selection process (requiring unusually high grades and test scores); they constitute a relatively homogeneous, high-level group of students. This fact may help explain the lack of significant correlation between prior knowledge and final course grade. Most students gave similar answers regarding their prior knowledge of various concepts, and the absence of significant differences in their answers may have contributed to a lack of significant correlation.
In any case, we found that students who had learned a procedural language such as Pascal or C were more successful than those who did not learn a previous programming language.

REFERENCES

[1] Barnes D. J. & Kolling M. *Objects First with Java: A Practical Introduction using Java*. Prentice Hall / Pearson Education.2003.

[2] Becker B. W. "Teaching CS1 with Karel the Robot in Java", *SIGCSE, The Proceeding of the Thirty-Second Technical Symposium on Computer Science Education*, 2001. pp 50-54.

[3] The Joint Task Force on Computing Curricula. Computing curricula 2001. *Journal of Educational Resources in Computing (JERIC)*, v1, n3 (Fall 2001).

[4] Wick M. R., Stevenson D. E. & Phillips A. T. "Seven Design Rules for Teaching Students Sound Encapsulation and Abstraction of Object Properties and Member Data", *SIGCSE, The Proceeding of the Thirty-Five Technical Symposium on Computer Science Education*, 2004, pp 100-104.