Performance in an Introductory Computer Programming Course as a Predictor of Future Success for Engineering and Computer Science Majors

Barbara T. Pioro
North Carolina A&T State University
pioro@ncat.edu

Kenneth A. Williams
North Carolina A&T State University
williams@ncat.edu

Abstract

In most schools, introductory computer programming courses are listed in the curriculum as required for computer science as well as all engineering majors. It is generally believed that the programming courses are not just about programming per se, but they provide a forum for teaching precise and logical thought processes. Thus, computer programming courses constitute not only a necessary background for the computer science majors by introducing basic concepts and techniques to be used and to be built upon in more advanced CS courses, but they also make a valuable contribution to the foundation of engineering education. The programming courses are thought to provide a valuable framework for the development of problem solving skills as well as creative thinking skills within any engineering major as well as computer science. It is commonly acknowledged that many students experience problems with computer programming classes. However, it is also acknowledged that success in a major is not determined by the student's ability to code textbook problems in the introductory computer programming class. The relationship between performance in a programming course and success in an engineering major or computer science has not been well established. In particular, a relationship between performance on different types of computing tasks and a major has not been sufficiently examined. In general, to the programming course, engineering majors bring different sets of experiences and expectations from those brought in by computer science majors; those factors have an effect on performance on tests and exams and manifest themselves in a variability of total scores and in a variability of scores on different types of computer programming problems.

In this study we examined scores obtained from the final exam questions in a computer programming course offered in the Spring of 2002. The scores were then reviewed in the context of students' records at the end of the Spring 2004 semester. The Spring 2002 programming tasks involved ten multiple choice questions and three programming problems. The problems were designed within the framework of the Rasmussen's skill-rule-knowledge model of human performance. Twenty five engineering and computer science students took the final exam. The data obtained in Spring 2002 were examined in view of the current students' performance and status. For each student, the analyzed data included scores obtained on individual exam problems, the student's self assessment scores for each exam problem, the GPAs from the Spring

2002 and Spring 2004 semesters, the major, as well as the retention and progress in the program.

The analyses of the Spring 2002 data identified a bimodal distribution of scores and showed significant differences in performance between the high-performing students and the low-performing students. This paper examines the predictive value of performance levels in introductory computer programming courses as indicators of overall student success in an engineering or computer science undergraduate program.