IDENTIFICATION OF NON-SUCCESS FACTORS IN A LARGE INTRODUCTORY COMPUTER SCIENCE COURSE AND CONSTRUCTIVE INTERVENTIONS FOR INCREASING STUDENT SUCCESS

Daniel K. Apple[1] and Donald Nelson[2]

Abstract *¾ A research design has been created for the WMU Introductory Computer Science Course that will leverage existing published research with classroom research to identify from student perspectives the key barriers to their own success. As these barriers are identified and articulated, there will be proactive changes made in the course material, a set of activities, use of assessment, and course structure to help the students increase their success. Success is being measured in several dimensions: obtaining an A, B, or C in the course; obtaining an A, B, C, in the required linked subsequent course; students' self-direct learning skills; their problem solving skills; and their ability to take on the perspective of and perform the role of a computer scientist. The paper will provide a list of the identified issues/barriers, and address these issues/barriers by applying at least two interventions. Also, there will be a detailed description of what was attempted and what the qualitative impact of these interventions was at the time of implementation, as well as a summative assessment on the whole course. A research plan for follow-on study will be discussed to open up more collaboration with other institutions, to develop new intervention strategies, and to redefine the structure of the course to continually increase the percentage of students who are successful.*

Index Terms *¾ Process Education™, collaborative learning, assessment.*

CS 111, Computer Science I, (CS1) is the introductory class for computer science majors at Western Michigan University. The current computer language is C++. In more recent years it has also taken on the role of a service course for numerous students in engineering, mathematics, science, and business information systems. Approximately 220 students enroll for the course each semester, and computer science majors make up less than one-half of the enrollment. It is a four-hour course consisting of large lectures (3 hours per week) and a laboratory that meets once a week for two hours. Lectures are designed to cover necessary concepts prior to a given laboratory session. Teaching assistants use time in lab also to cover or review topics, but the majority of laboratory time is allocated to hands-on computer work. In the laboratory there is one workstation per student. Material is structured and covered as though this is a first programming experience for students; although, they actually bring with them a wide range of computer programming skills.

The success rate, as measured by the number of students who obtain a grade of C or better, is low, often less than 50%. For too many students who take this class, the large lecture and laboratory setting simply is not working. Our goal in this work over the next several semesters is to identify problems that are inherent in the course and problems that students bring to the course, and then attempt to help alleviate these problems with appropriate actions, employing the philosophy of Process Education ™, "an educational philosophy focusing on improving students' learning skills (in cognitive, social, affective, and psychomotor domains) and creating 'self-growers'". (See http://www.pcrest.com/pe.htm.) In addition to the improvement of success rates in the class, our hope is to equip students with learning techniques and behaviors that enable them to be self-growers and increase their probability of success in future courses, where the structure planned for this course won't necessarily be available. As this is basically a freshmen class, catching students early, particularly computer science majors, and exposing them to the planned techniques and activities should increase their overall success rate in their programs of study.

Past experience has shown that students in this class have behaviors or traits that increase risk factors. Many are experiencing university life for the first time and haven't developed an appropriate sense of their own responsibility for, or ownership of, their learning. Absences from lectures are usually high, often exceeding 50%. Many students have not acquired the ability to read and comprehend highly technical material and do not read the textbook portions specified on the course syllabus. In large lecture classes, students often take a passive attitude toward the instruction and learning process, and fail to recognize areas of incomplete understanding. Many lack the confidence to succeed in the course and also lack the affective skills to deal with frustrations that result when attempts are made to design and write computer programs. Often language skills have not developed to a level necessary to deal with understanding logical and syntactical concepts needed for communicating with a compiler. These factors lead to early frustration and disengagement on the part of the students.

[1] Daniel K. Apple, Pacific Crest, dan@pcrest.com
[2] Donald Nelson, Western Michigan University, nelson@cs.wmich.edu

The course itself possesses a number of risk factors for students. It requires a style of logical thinking that many students have not experienced upon entry to the course. The content is hierarchical in nature, so failure to grasp early concepts poses serious problems for mastery of subsequent concepts, and this combined with passive behaviors in lecture often results in failure to provide timely warning signs of impending trouble. The large lecture setting enables passive learning behaviors, absenteeism, and disengagement.

Particular tools that have been or will be employed to address these problems include collaborative learning activities, assessment, and development of techniques to improve reading. In addition, the size of the section in question has been reduced to 80, and it was offered in a new high-technology classroom, which readily enables team activities in teams of size four or five. During the first semester of implementation, students were involved in team activities; however, these were employed in fewer than half the available class meetings. Ongoing assessment activities for students were not done.

For future offerings, team activities will be devised for nearly all of the class periods throughout the semester, the aim of these being to keep students more actively involved in the course throughout the semester and to learn through collaboration.

Assessment activities will be designed to provide frequent feedback and continuous improvement as students progress through the course. Students will assess themselves and each other. Students and instructor will be involved with assessment of the course as the term progresses.

Students will be required to keep reading logs that are structured to provide additional direction in the reading of technical material.

Evaluation of individual student performance will be done in much the same manner as in the past. There will be examinations, individual programming assignments done through the laboratory, and individual classroom assignments. An added component will be teamwork, which will count at most 15% of the overall grade. Conducting of the laboratory will remain the same as in the past, since each laboratory section will contain students from all lecture sections, not just the lecture section using Process Education™ philosophy.