4. Project Description.

Prior NSF Support: None of the investigators has received prior NSF support.

a) Goals and Objectives: This project has two goals. First, we aim to improve the retention rate in the CS program at our institution and at comparable regional colleges and universities, particularly among minority, female, and rural students. Second, we will enhance the ability of less selective institutions to include a strong foundation in mathematics in their undergraduate CS programs. To reach these goals, the project will:

- Adapt and implement successful pedagogical practices and materials from CS and mathematics education and asynchronous learning networks.
- Encourage retention among second and third year CS majors by offering them an active role in the project. We will train peer mentors to staff discussion lists and help sessions. Student assistants will help implement and maintain the web-based software. We will encourage the students to make presentations on their work at departmental colloquia, at regional professional conferences, and at the Georgia Academy of Sciences.
- During project development, use focus groups of students from targeted populations for formative qualitative and quantitative evaluation of the practices and materials.
- Assess outcomes by evaluating performance, attitudes, and retention rates of students who use the practices and materials and of a control group, noting especially the effects on the targeted student populations.
- Incorporate the results of evaluations into the project’s ongoing development.

- Conduct faculty development workshops, at first internally and, later, outside our institution, to demonstrate our practices and materials, to get feedback on their quality and usefulness, and to foster their collaborative development.
- As evaluation shows what is and is not successful, incorporate the practices and materials into all sections of CS 1 and 2 at our institution, and encourage their use in CS curricula at comparable schools.
- Build on the project to develop materials for the asynchronous delivery of lessons on the mathematical foundations of CS to support a rigorous treatment of more advanced coursework in data structures, programming languages, algorithm analysis, and computer organization and architecture.
- Disseminate the results of the project through faculty development workshops and tutorials, presentations at professional conferences, articles in computer science education and asynchronous learning journals, and a project web site.

These goals fit well with the objectives of the Department of Computer Science at the State University of West Georgia (UWG), of UWG, and of comparable departments. We have been very concerned with low retention rates. Nationally, some fifty percent of students who express a desire to major in CS fail to complete the introductory sequence. Our situation is even worse. We have a failure or withdrawal rate of more than fifty percent in each of the first three courses in our program. The drop rates for minority, female, and rural students is higher still. Though many factors contribute to these poor retention rates, weak mathematical skills without doubt play a leading role.

In addition, we, like many others, are reviewing our curriculum in light of the revised criteria of the Computer Science Accrediting Board. We have also followed the development of the ACM/IEEE Computing Curricula 2001, including the release of the "Strawman" draft proposal and the discussion of it at the recent ACM SIGCSE Symposium. The emphasis placed in both documents on the importance of the mathematical foundations of CS has focused our attention upon their role in our curriculum and on our students' difficulties with foundational concepts.

UWG and the Department of Computer Science: UWG is a coeducational state supported regional university with a current enrollment of almost 8800 students, roughly 75% attending full time and 25% part-time. The undergraduate student body of some 6800 students is a diverse group, with 20% African-American and 65% female. The majority of our students are first-generation college students from blue-collar backgrounds. The typical full-time student works twenty hours per week, and 16% are older than 25. The middle fifty percent of scores on the SAT I are from 400 - 500 on both Mathematics and Verbal tests.

UWG's primary service area, a 37-county region of rural counties and small cities, was the site of a large textile industry, with much of the population doing piecework for the mills. With the removal of most of the mills to foreign countries, economic change has been rapid, with many workers shifting to service and construction jobs in the Atlanta Metropolitan area. Public education in the region has not, until recent years, enjoyed adequate support. Most high schools in our service area rank in the third and fourth quartiles in state rankings of standardized test scores.

The state of Georgia, local Chambers of Commerce and governments, and the University are working together to raise educational standards in the West Georgia region to attract better employers and to be able to provide an educated workforce for them. The

CS Department was accordingly selected in a University-wide planning process as a center for enhancement and growth. Since splitting from the Department of Mathematics five years ago, our faculty has grown from three to nine full-time faculty members, five with doctoral degrees in CS, one with a doctorate in applied mathematics. We offer the Bachelor of Science in Computer Science, with concentrations in Computer Science and Software Engineering. We expect to receive approval from the University System of Georgia for a Master of Science in Applied Computer Science. In spring, 2001, the department will move into a new Technology Enhanced Learning Center, which will give us the room to expand our laboratory facilities beyond the one seventeen station laboratory we now have.

The quality of our program has increased as we have brought the curriculum into accord with national standards, built stronger ties with local industry, and received strong institutional support. This improved quality is demonstrated by the strong placement and career advancement record of our graduates, but has come in part at the expense of a high dropout rate. Some 275 - 300 students enroll in the first course in the introductory sequence each year, but only about 120 in the second half of the sequence and 40 in the second-year sequence.

The Computer Science Department can best improve both its quality and retention rates by recognizing and meeting the particular needs of our current students and of prospective students from our service area. Since we are not one of our state's flagship institutions, we, like comparable regional institutions, cannot fulfill our mission by attracting a large number of well-prepared students from affluent suburban high schools.

We must instead use efforts like this project to help the region we serve by helping our students overcome their educational and economic disadvantages.

b) Detailed Project Plan: This project will adapt to the requirements of less selective regional institutions the efforts of Alfred Aho and Jeffrey Ullman, David Gries, and the Math-Thinking working group [MTWG] to increase the role of mathematical foundations in undergraduate CS. In adapting their work, we will apply strategies useful with less well-prepared students as described in recent pedagogical research. Our goals are to improve retention rates and to strengthen our curricula.

Aho and Ullman provided early stimulus for this project with their text, *Foundations of Computer Science* [Aho92, Aho95]. It integrates the topics taught in traditional data structures and discrete mathematics courses. In 1993, the CS program at UWG replaced separate courses in those areas with a two-term sequence organized around Aho and Ullman's work. The first course covers propositional logic, mathematical induction, recursion and recurrence relations, and combinatorics, together with their application to linear data models. The second extends the treatment to nonlinear data models, finite state automata, grammars, and first-order predicate logic.

The department has been very happy with the results of this course sequence. Students leave it with a much firmer grasp of the use of concepts from discrete mathematics than when the concepts were introduced in the separate discrete mathematics class. We have recognized, though, that our students have great difficulty making the transition from our introductory sequence, with its focus on software development, to the more rigorous approach of this sequence. Not surprisingly, this seems especially true for those whose mathematics background is weakest, which includes many of our minority, female, and rural students.

We do not want to water down the mathematical content of our upper-level courses, so we have looked for ways to increase the mathematical rigor of the introductory sequence. David Gries [Gries81, Gries94, Gries96] has shown how mathematics can be a tool for increased understanding rather than merely an add-on to an already crowded curriculum. Doug Baldwin, of the Math-Thinking working group, has pioneered at SUNY at Geneseo the development of mathematically rigorous introductory CS courses that pay close attention to making mathematics meaningful to beginning students. [Baldwin 94, Baldwin 96, Tomer00]

Unfortunately, the differences between the mathematical preparation of the students taught by Baldwin and Gries and that of our students makes straightforward application of their ideas problematic. (For example, their entering students have average SAT Mathematics scores from 150 to 250 points higher than the average at UWG.) We can adopt the spirit of their work, but must adapt their materials to suit our environment.

Our thinking in planning this adaptation has been strongly influenced by the *Crossroads in Mathematics: Standards for Introductory College Mathematics Before Calculus* [Cohen 95]. This report was prepared for the American Mathematical Association of Two-Year Colleges and funded by the NSF. It describes policies and strategies for teaching college-level mathematics to students like those we target: weak in mathematical preparation, employed while attending college, and/or lacking a family history in post-secondary education. Its "Standards for Pedagogy" section advocates a generally constructivist approach by endorsing strategies that provide for student activity and student-constructed knowledge. The standards also, however, recognize the importance of the teacher's role in structuring the student's experience, and of the time that the student spends on appropriate mathematical activities.

The report recommends using technology as a tool for doing mathematics and as a medium for instruction. It encourages building connections between mathematics and other experiences and disciplines. It stresses the importance of using multiple instructional strategies including collaborative and interactive learning; numerical, graphical, symbolic, and verbal modeling; data collection and analysis; and multi-step problem solving.

In addition to the *Crossroads* report, we will draw from recent pedagogical research in CS that advocates: frequent practice [Schollmeyer96, Tilbury97]; self-paced learning [Liu96]; active learning [McConnell95]; constructivist techniques [Hadjerrouit99, Ben-Ari98]; incremental learning and attention to students' motivations and backgrounds [Buck00]; and encouraging student reflection [Fekete00].

These recommendations challenge us to adapt practices and materials for asynchronous, on-line delivery that go beyond short-answer and multiple choice questions. We will use the resources offered by WebCT [WebCT], the web-based course development and learning environment. WebCT provides facilities for the presentation of hypermedia content; on-line quizzes; threaded discussion lists; chat rooms; white boards; public and private e-mail; and group discussions and presentations. It allows the instructor to monitor student participation and progress, and to conduct student surveys. We can extend WebCT to provide additional features, like animations and simulations, by including hyperlinks to them from our WebCT site.

By using WebCT, we will avoid the "Lone Ranger" development syndrome warned against in the NSF-DUE's *Shaping the Future* [NRC96]. We will not have to create yet another on-line learning environment that would probably find only limited external use. We can instead join an established community of course designers with whom we can share ideas and expertise. WebCT.com, the application's associated web site, sponsors faculty discussion groups and user conferences, posts articles on on-line teaching and learning, and provides links to other web sites. Our materials will thus be in a format with which many others will already be familiar, making their adoption and collaborative extension more likely.

Using WebCT will let us focus on content and pedagogy. The most important mathematical skills for the first year of undergraduate study in computer science are drawn from algebra and discrete mathematics. Topics include sets, relations and functions, linear equations and inequalities, Boolean algebra, logical reasoning, recursion, mathematical induction, number systems, matrices, asymptotic notation, and trees. [Aho95, Kelemen99, Cortina00]

Our approach to this material will be consistent with the pedagogical research previously cited: (a) Introduce mathematical terminology and concepts "just in time." (b) Leave for later courses those aspects of mathematics whose application is not useful in introductory CS. (c) Review relevant concepts from high school mathematics to connect new ideas with students' prior understanding. (d) Discuss concepts at a level of abstraction appropriate for the targeted students, using simulations, visualizations, metaphors, and instructional techniques like scaffolding [Hogan97] to structure the student's experience to make challenging ideas more comprehensible. (e) Introduce topics in easily understandable increments, permitting the assimilation of one facet of a concept before the next facet is introduced. (f) Have the student interact with the material and get feedback as it is presented. Never expect the student to receive information passively for more than about two minutes before providing interaction. (g) Stress the use of mathematics as a tool for understanding computer systems and software development. Motivate each concept by showing how it can make what the students study in CS 1 and 2 more understandable. (h) Encourage students to actively construct their understanding by solving problems rather than simply by performing calculations. Provide hints, alternate solutions, and links to related material as appropriate. (i) Encourage collaboration by forming small groups to work on problems and report on their results. (j) Integrate previously learned material into new lessons to help students integrate the old material into their conceptual framework and use it as a foundation for constructing new understanding. (k) Allow flexible paths through each lesson to suit individual needs. Provide students who need extra time to master a concept both additional on-line activities and face-to-face tutoring with a peer mentor. (l) Give feedback that stimulates reflection and self-assessment.

Project Activities: We will develop twenty-four asynchronous on-line mathematics lessons to support the introductory CS sequence, with twelve for each course. We will assume that the sequence uses an object-oriented development methodology, with Java or C++ as the programming language. We will also assume that the first course is taught using an "objects early" approach, but will not tie the lessons to a particular text. We will provide multiple paths through the lessons to accommodate instructor preferences in early or late treatment of recursion, inheritance, and streams.

Each lesson will be designed to take a typical student 40 to 60 minutes to complete. The material will be structured to allow well-prepared students to work through a lesson more quickly, while the less prepared will be able to spend more time on additional activities. Each lesson will be broken into sections to let students set their pace individually. At least three lessons for each course will have a collaborative component that requires a small group of students to work together to reach a solution.

For example, a first lesson about Boolean expressions could be scheduled to coincide with lectures and laboratory exercises about selection statements. The concept of a true/false condition would be introduced, with examples and questions drawn from real-world experiences. Then, the lesson would review numeric expressions, with a few exercises for practice. The concept of an expression could now be extended to incorporate true/false conditions, with an explanation of the Boolean type. The examples used earlier in the lesson would now be restated in pseudocode. Exercises including instances of both numeric and Boolean expressions would follow to strengthen the student's understanding of an expression, with explorations of their value and use in software development. Finally, Boolean operators would be introduced and used in Boolean expressions with two sub-expressions, with exercises on the operators followed by practice with both simple and compound Boolean expressions.

We will support the lessons with materials for both faculty members and peer mentors. Peer mentors will be provided with lesson objectives, hints on explaining the concepts, and exercises for extra face-to-face practice. For faculty members, we will provide an on-line instructor's manual, with suggested paths through the lessons and instructions for adding new material. The project web site will also include a threaded discussion list and a frequently asked questions page.

The project time line is as follows. During 2001, we will develop the lessons to accompany CS 1 during the summer and the lessons for CS 2 during the fall. We will run a workshop for the faculty members teaching CS 1 and train the first group of peer mentors in August. We will introduce the lessons in selected sessions of CS 1 and 2 during the 2001 - 2002 academic year, and evaluate preliminary results and make indicated changes during the summer and fall of 2002. If initial results are promising, we will use the lessons in all sections of the introductory sequence by the spring semester of 2002. Dissemination of results will begin in the spring of 2002, with wider dissemination later that year and during 2003.

Facilities and Resources

Equipment Request: The proposed activities can be accomplished by equipping a laboratory of nineteen networked desktop computers. Seventeen of the computers will be for students enrolled in CS 1 and 2, with one for a peer mentor and one for a student assistant developing and maintaining the software. With 200 - 240 students in the introductory sequence each semester, this will provide the students with adequate computer access for both individual and group work on the lessons and for mentor-led help sessions. A laser printer, scanner, and data/video projector will provide support for data collection, presentations, and group work.

Networked multimedia personal computers (PCs) running Windows NT will be necessary for the purposes of this project. The machines should use a Pentium III 667MHz processor, and have 128 megabytes of RAM and a ten-gigabyte hard drive. The need for extensive multimedia access dictates the use of at least nineteen-inch monitors, DVD drives, and sound cards. We will also need a workgroup server and hub.

Equipment on Hand for the Project: The Computer Science Department has primary use of a seventeen-station networked PC laboratory. The computers dual-boot Windows and Linux, and are heavily used for closed and open laboratories. The department hopes to obtain corporate or state funding for a thirty-eight station laboratory when it moves to the new Technology Enhanced Learning Center. Its students may also use the sixty-station Arts and Sciences laboratory, though that is usually crowded with students from other departments. Each faculty member in the department has a desktop office PC. The office and laboratory computers are maintained by the technical support staff of the College of Arts and Sciences, with backup as necessary from the University's Department of Information Technology Services (ITS). ITS also provides two Sun workstations for faculty and student use, plus campus web and mail servers.

c) Experience and Capability of the Principal Investigators: Dr. Will Lloyd, founding chair of the CS Department from 1994 - 1998, revised the syllabi for CS 1 and 2 to emphasize object-oriented development. He developed the Data Structures and Discrete Mathematics sequence to replace the formerly separate Data Structures and Discrete Mathematics courses. He also designed and taught the first Theory of Computation and Software Engineering courses offered at UWG. He has led faculty development workshops for University System faculty on C++ and Java programming and on undergraduate CS curriculum development. He has used WebCT in courses for two years, and, as chair of the UWG Technology Planning committee and member of the

Distance Learning committee, develops university policy on distance education. His industrial experience includes work as an applications programmer and project manager.

Dr. Rebecca Rizzo has been a mathematics instructor at the university, community college, and junior high school levels. At the college level, she has taught remedial mathematics, college algebra, Mathematica, statistics, precalculus, calculus, differential equations, and mathematics for elementary teachers. At the junior high school level, she taught seventh and ninth grade mathematics and was co-developer of a computer laboratory for technology instruction.

Dr. Adel Abunawass helped develop the Principles of Computer Science 1 and 2 sequence at Western Illinois University. This sequence offered a breadth-first introduction to computer science to accompany a co-requisite sequence in programming. He has published extensively on neural networks, software engineering, and computer science education, and has professional experience as a software engineer and as a consultant on applications of artificial intelligence.

d) Evaluation Plan: The primary criteria for evaluation of the project will be its effect on retention rates and its impact on our ability to strengthen the role of the mathematical foundation of CS in our undergraduate curriculum. To assess these outcomes, we will perform both formative and summative evaluations. The results of these evaluations will shape the on-going development and maintenance of the lessons, and guide us in the development of materials for upper-level classes.

During the development of the lessons, we will hold meetings with participating faculty members and targeted students to do qualitative evaluation of the material. We will also solicit the views of faculty members in the Departments of Mathematics, Curriculum and Instruction, and Media and Instructional Technology at UWG. We will use WebCT to conduct on-line quantitative surveys of student attitudes and to monitor student activity.

When the lessons have been implemented, we will compare the retention rates of students in sections that use the lessons and in those that do not. We will analyze the rates of those who pass CS 1, those who continue from into CS 2, those who pass Data Structures and Discrete Mathematics 1, and those who graduate from the program. For the sections in which the lessons are used, we will analyze the correlation between use of the material and performance. We will conduct qualitative and quantitative evaluations of upper-level students and alumni to assess the impact of the lessons on the quality of their undergraduate educational experience.

e) Dissemination: Dissemination of our experiences will begin locally with departmental colloquia and workshops for faculty teaching CS 1 and 2. We will then make presentations to colleagues in other departments at brown-bag lunches and colloquia and at the annual university-wide Celebration of Scholarship. We should find interest in the science and mathematics departments, in the Department of Management and Business Systems, and in the College of Education. We expect these local venues to foster exchanges of ideas and opportunities for future collaboration.

Within the University System of Georgia, we will hold a faculty development workshop at a yearly meeting of the Academic Advisory Committee for Computer Science and give a presentation at the University System Annual Computing Conference. Most of the fifteen regional universities and colleges and fifteen two-year colleges of the

University System face similar retention and curricular problems as UWG. We hope to identify several institutions that will be interested in using the materials produced in this project or participating in its further development.

Farther afield, we should find interested audiences at regional conferences of the ACM and the Consortium for Computing in Small Colleges. We will send submissions to conferences like the SIGCSE Technical Symposium on Computer Science Education, the SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, the International Conference on Asynchronous Learning Networks, and the National Education Computing Conference. We will also submit papers for publication in the *SIGCSE Bulletin*, the *Journal of Asynchronous Learning Networks*, and/or the *Computer Science Education Journal*.

All lessons and supporting materials will be available at the project web site. We will also make available our evaluation instruments and results so that others may build upon our experience.