Teaching Excellence Statement of the Computer and Information Science (CIS) Department

Introduction

The CIS Department does well with respect to most of the criteria specified in the call for nominations for the 2000-2001 University Departmental Teaching Excellence Awards. However, rather than trying to address each criterion to the same extent, we focus our attention in this statement on two items in which we excel[1]. The first is an innovative three-quarter sequence of courses developed by faculty in our department; students intending to major in CIS (Computer and Information Science via the College of Arts and Science), or CSE (Computer Science and Engineering via the College of Engineering) are required to start by taking this sequence. The second is an innovative set of assessment and feedback mechanisms we have developed and implemented to continuously monitor and improve our program. These two items are the topics of Sections A and B respectively. In Section C, we briefly address the remaining criteria specified in the call for nominations.

A. Innovative Sequence on Software Engineering

A.1 History

In 1996, the Department began an ambitious curriculum development effort. The overarching goal of this project was to improve the ability of computer science graduates to design and develop high-quality, reliable software. To this end, we focused on improving the quality of the introductory sequence in computer programming (software development) by:

- Promoting a specific view of software, of how to design it, and of how to build it.
- Introducing a discipline of software development based on sound and time-tested principles.
- Developing and using a suite of tools that promote the desired mental model of software and that assist students as they engage in a disciplined approach to software development.
- Adding intellectual content to a course sequence traditionally dominated by skill development.

The result of this effort is a new introductory-level three-course sequence in software design and development. The three courses are:

- CIS 221: Software Development Using Components
- CIS 222: Development of Software Components
- CIS 321: Design and Analysis of Software Components.

These courses are based on the view that software systems should be conceptualized and realized as a composition of individual components. Further, to the extent possible, this should be accomplished using available, off-the-shelf components. This view reflects a bias towards engineering, and, indeed, our course sequence is founded on the twin pillars of all component-based engineering: system thinking and mathematical modeling.

A.2 Motivation

In our opinion, traditional introductions to software development emphasize a build-from-scratch mentality from which students form a conceptual view of software, and of how to build it, that is contrary to good software-development practice. Further, and perhaps more importantly, we believe that it is often impossible to identify an intellectual foundation for the content of traditional programming courses. Rather, the

[1] We should also note that because of space limitations, we restrict attention in this statement to our undergraduate programs.

emphasis seems to be on programming-language-specific skill development. This is clearly problematic, if for no other reason than the fact that programming languages are constantly changing. How will our students maintain competence in the face of this change? We believe the answer lies in a deeper understanding of time-tested and time-proven principles that transcend programming language particulars. In our opinion, the skills-development approach to introductory programming sacrifices consideration of such principles and of the deeper intellectual foundations upon which they rest.

A.3 Realizing Project Goals

The following summary explains how we have attempted to realize project goals in the course sequence.

Goal 1: *Promoting a specific view of software, of how to design it, and of how to build it.*

Starting in CIS 221, students are supplied with a catalog of software components. In all programming assignments students are instructed to use certain catalog components to solve the problem at hand. Thus, from the beginning, students use catalog components as a matter of course and do not engage in build-it-from-scratch programming. The first half of CIS 222 continues this same emphasis. Then, in the second half of CIS 222, students begin to look inside the provided catalog components to see how they are put together, and they complete a series of assignments where they build catalog components. Often students see that catalog components in turn are built out of other components. In CIS 321, students work on a course-long project developing a fairly substantial software system using both existing catalog components and newly introduced components. At no point in the curriculum do we show students software or ask students to build software that is not component based.

Goal 2: *Introducing a particular discipline of software development based on sound, time-tested principles.*

The entire course sequence uses a discipline based on proven principles such as system thinking, mathematical modeling, modular reasoning, information hiding, abstraction, and substitutability, for example. We have taken great care to make sure that all software artifacts, used in the course sequence, overtly reflect these principles in both their conceptual design and in their physical manifestation. Hence, the manifestation of these principles is explicit in all components that students use and/or build. Further, we employ a discipline of practice to which students must adhere. The discipline prohibits practices that are known to result in software artifacts violating underlying principles. Thus, students learn to work within a discipline that is rooted firmly in sound and proven principles.

Goal 3: *Developing and using a suite of computer tools that promote the desired mental model of software and that assist students as they engage in a disciplined approach to software development.*

We refer to our tool suite as the workbench. Through its interface and through the operations it allows, the workbench actively promotes a component-based view of software and of its construction. The workbench interface explicitly features catalogs of components. It allows users to create, copy, move, and delete catalogs, and to display appropriately structured listings of components within catalogs. It similarly allows users to create, copy, delete, view, and edit components within catalogs. The traditional view of software components being files within a computer's file system is not supported by the workbench.

Goal 4: *Adding intellectual content to a course sequence traditionally dominated by skill development.*

Basing a course sequence in software development on a specific framework for thinking about and building software allows, in a quite natural way, for the introduction of intellectual content. Course materials and in-class presentations include explicit discussions of the framework's underlying principles and of their realization in software. The goodness of software no longer is strictly in the eye of the beholder, even at the introductory level. Rather, it is in the degree to which software manifests deeper principles. Further, points of discipline can be justified by appeal to principles. Relative to our particular approach to software, time-tested engineering principles become relevant in the software development process, and become central to what students learn and do. Examples include:

- Understanding dependencies between components in a system: what they are, what they mean, why they are important, which dependencies need to be specified, and how to do it.
- Minimizing component-to-component dependencies: why this is important, and how to do it.

- Achieving generality through parameterization: why this makes off-the-shelf components more useful, and how to do it.
- Achieving modular-system design: how this impacts maintenance and promotes interchangeable components, and how to do it.

Thus introductory courses in software development can and do become rich in the intellectual content, concepts, and principles of engineering design.

A.4 Course Pedagogy

CIS 221: This course makes exclusive use of active learning techniques. The material that would normally be delivered in lecture has been written as individual reading units. As homework for each class, students read a unit of material and answer questions embedded in the reading as comprehension checks. Upon arriving in class, students work in fixed activity teams of four students each. Together they do a series of pencil-and-paper exercises designed to deepen understanding of the material and to improve their skills. The instructor and grader circulate through the classroom interacting with the activity teams as appropriate. Thus, as students are engaged in those activities that expand their comprehension and improve their skills, they have the advantage of working directly with other students and with the instructor.

It is interesting to contrast this pedagogical approach with traditional lectures. In a lecture format, students listen to a lecture, take notes, and ask questions in class. Then, using notes from class and a textbook, they work individually on homework problems to deepen understanding and improve skills. The key difference, then, between the two approaches are the circumstances under which, and resources available when, students are actively engaged in comprehension and skill building exercises.

CIS 222 and 321: These courses are lecture-based. However, activities are woven into the lecture format and students do work on many in-class activities with their neighbors.

The following table summarizes our pedagogical techniques:

	CIS 221	CIS 222	CIS 321
activity-centered classroom	√		
lecture-centered classroom/integrated activities		√	√
two-person closed-lab teams	√	√	√
two-person project teams			√

Finally, we make liberal use of manipulatives in all three courses. These include such things as plastic stacking cups that toddlers play with, Lego blocks, PVC pipes, and even people inside of refrigerator boxes! We use these manipulatives to help students

- form mental models of mathematical concepts
- form mental models of component behavior
- understand algorithms

The use of manipulatives is quite popular with students.

A.5 Further Details

The web-page for the new course sequence is at http://www.cis.ohio-state.edu/~weide/sce/now. This site includes links to publications concerning the course sequence and to related awards. The awards include

- 1997: CIS Department Excellence[2] in Teaching Award (Chris Jermaine)
- 1998: CIS Department Excellence in Teaching Award (Tim Long and Bruce Weide)
- 1999: CIS Department Excellence in Teaching Award (David Mathias)
- 2000: CIS Department Excellence in Teaching Award (Steve Fridella)

[2] The Department gives annual awards recognizing excellent achievements of its students, excellent teaching by instructors, and excellent service by faculty and staff.

- 2000: OSU College of Engineering Ralph L. Boyer Award for Excellence in Teaching Innovation (Tim Long and Bruce Weide): "For outstanding contributions to the improvement of the undergraduate engineering curriculum with the development of a new paradigm for introductory programming courses and innovative instructional techniques."
- 2000: IEEE Computer Society Computer Science and Engineering Undergraduate Teaching Award (Tim Long and Bruce Weide): "For innovative work in the content and pedagogy of introductory computer science education, linking research advances in software engineering with educational delivery of the material taught in the introductory courses."

Finally, we note that this effort was supported by two grants from the National Science Foundation and by a grant from the Fund for the Improvement of Postsecondary Education from the Department of Education.

B. Self-Evaluation of Teaching and Learning Quality

During the past three years, we have developed and implemented a number of assessment and feedback mechanisms to monitor and improve the quality of our programs. During the recent accreditation evaluation by *ABET* (Accreditation Board for Engineering and Technology) and by *CSAB* (Computing Sciences Accreditation Board) of our CSE program, these mechanisms were rated very highly. Indeed, our approach is considered an excellent model for Computer Science and Engineering programs and is recommended as such by CSAB to programs interested in developing such mechanisms (see http://www.csab.org/obj_and_assess-ppt/index.htm). In this section we summarize these mechanisms, and sketch some of resulting improvements in our undergraduate programs[3].

In order to assess the effectiveness of any program, it is important to precisely define the *objectives* and the expected *outcomes* of the program; only then can we develop suitable assessment instruments to measure the achievement of these objectives and outcomes. In our program, we first formulated precise objectives, and, for each objective, a set of related outcomes; once this was done, we identified how various aspects of our curriculum were related to each objective and outcome; and developed a variety of assessment tools to measure how well we were achieving each of our objectives and outcomes. We also put into place well-defined processes to ensure that the results of the assessments are used in an on-going manner to improve the effectiveness of the program. In all of this, a key focus of our activities is to ensure the involvement of all our main constituents.

In Section B.1, we outline the processes related to our objectives and outcomes and briefly discuss a subset of the objectives and outcomes. In B.1, we also discuss how various components of our curriculum relate to the different objectives and outcomes. In Section B.2, we summarize the main assessment instruments we have developed to measure the effectiveness of our program. In B.2 we also describe the feedback mechanisms we use to integrate the inputs obtained from our assessment instruments and to identify possible improvements in the program, based on these inputs. In Section B.3, we summarize some of these improvements.

B.1 Objectives and Outcomes

The current set of educational objectives and the related outcomes of our undergraduate programs are published on the departmental web pages; see the 'new brochure' accessible at

http://www.cis.ohio-state.edu/Academic-Programs/Undergraduate/index.html

B.1.1 Process for Determining Objectives

The original set of objectives and outcomes was arrived at following extensive discussions in our Undergraduate Studies Committee (UGSC) that includes faculty (both tenure-track and non-tenure track faculty), student representatives, and the department's professional undergraduate advisors. Following these discussions, a draft set of objectives was posted on student and faculty newsgroups for their comments, discussed at an open student meeting as well as at a faculty meeting, and presented to our industrial advisory board

[3] Only our CSE program was evaluated by ABET and CSAB. Nevertheless, because of the close correspondence between our CIS and CSE programs, these mechanisms and the resulting improvements apply to both.

for their comments. A final set of objectives and outcomes was drafted based on all these discussions, and approved by the CIS faculty in the fall of 1997.

Since then we have obtained feedback on the appropriateness of these objectives and outcomes from various constituencies via the processes described in Section B.2. UGSC discusses this feedback on a regular basis (typically during its meetings in spring quarter). The feedback we have received has been very positive; based on the feedback received during the last three years, UGSC proposed some minor structural changes in the statement of objectives and outcomes. The revised set of objectives and outcomes was again discussed by students, the industrial advisory board, and faculty, and approved by faculty in the fall of 2000. The set of objectives and outcomes that appears on the web pages cited above is this revised set; for comparison, the original set is also accessible from the same pages. For the future, we plan to continue fine-tuning our objectives and outcomes, based on the feedback from constituencies, as well as in response to changes in the computing discipline.

B.1.2 Objectives and Outcomes: Some Details

One of our objectives reads as follows:

> To provide graduates with a thorough grounding in the key principles and practices of computing, and in the basic engineering, mathematical, and scientific principles that underpin them.

Corresponding to this objective, we have identified three outcomes, these being:

- Students will demonstrate proficiency in the areas of software design and development, algorithms, operating systems, programming languages, and computer architecture.
- Students will demonstrate proficiency in relevant aspects of mathematics, including discrete mathematics and probability, as well as electrical circuits and devices.
- Students will successfully apply these principles and practices to a variety of problems.

Thus while the objective provides a high level statement of a key program goal, the corresponding outcomes list specific proficiencies that students will acquire in order to ensure the achievement of the objective. Another of our objectives reads:

> To provide students with appropriate social and organizational skills.

The outcomes corresponding to this objective are:

- Students will demonstrate an ability to work effectively in teams.
- Students will demonstrate an ability to communicate effectively.

It is perhaps worth noting that, in our experience and based on the results of the assessments described in B.2, these are among the most difficult outcomes to achieve. Nevertheless, the input we have received from several constituencies including alumni, employers of our graduates, our industrial advisory board, etc., stress the importance of these outcomes. Some of the changes we have made recently in the curriculum, and described in B.3, were dictated by exactly these considerations.

B.1.3 Relation Between Objectives & Outcomes and the Curriculum

Ideally, it would perhaps be best to first design the objectives and outcomes of a program, and then design the curriculum based on the objectives and outcomes. Clearly we could not do this since we already had a full curriculum in place long before we drafted the first statement of objectives and outcomes for our undergraduate programs. What we did therefore was to ask, for each course in our curriculum, the group of faculty members involved in that particular course for their consensus evaluation of what contribution that particular course makes to each of our objectives and outcomes. After discussions in UGSC, we agreed upon the following precisely-defined scale to specify these contributions, and all faculty used this scale in their evaluation of their own specific courses:

- *XXX* means the substance of the particular program outcome is a primary theme of the course; a significant fraction of course time (perhaps 3 weeks or more, often woven through the fabric of the course) is directly related to this outcome.

- *XX* means the substance of the particular outcome is a secondary theme of the course; a smaller fraction of course time (perhaps 1-2 weeks) is directly related to it.
- *X* means the substance of the particular outcome is not a theme of the course, but it is still treated in the course a non-trivial way; a smaller fraction of course time (perhaps 1-2 hours) is directly related to it.

The complete table of our courses and their contribution, measured using the above scale, to the various outcomes is available on our web site. A small portion of this table is reproduced below; this portion concerns the contributions made by CIS 221, 222, and 321, the three courses discussed in Section A.

Course	**1a**	**1b**	**1c**	**2a**	**2b**	**2c**	**3a**	**3b**	**4a**	**4b**	**5a**	**5b**	**5c**
CIS 221	XXX	XX	XXX	X				X	X	X	X	X	X
CIS 222	XXX	XX	XXX					X	X	X	X	X	X
CIS 321	XXX	XX	XXX					X	XX	X	X	X	X

The columns headed (1a), (1b), (1c) refer to the three outcomes corresponding to objective number 1, concerning providing graduates with a thorough grounding in principles and practices of computing; similarly for the other columns. The values under (1a), (1b), (1c), for these courses makes it clear that the primary contribution of these courses is toward meeting these outcomes. Outcome (2b) states that "[students will] demonstrate an understanding of the basic principles of physics and at least one other laboratory-based discipline." Again the table makes it clear that these three courses do not contribute toward meeting this outcome. Outcome (4a) has to do with team-working skills; thus, according to the table, CIS 321 (because of its team-based project,) makes a stronger contribution toward this outcome than do the other two courses.

Having a clear statement of the relation between the courses in the curriculum and the various program objectives and outcomes allows us to respond to constituent concerns about specific outcomes in a well-defined manner. Thus, for example, in response to concerns expressed by alumni about their preparation with respect to oral communication skills, the faculty members involved with CIS 601, the most important course –according to our table– with respect to this outcome, proposed suitable changes to this course.

B.2 Assessment and Feedback Mechanisms

We have set up assessment mechanisms to ensure that we receive input from all our important constituents, in particular, current students, alumni, and employers of our graduates. In this section, we briefly describe each of these mechanisms, and then discuss the feedback mechanisms that we have instituted to integrate all of this input and identify strengths and weaknesses and possible improvements to our program. Note that we omit descriptions of standard assessment mechanisms such as *SEI*s (Student Evaluation of Instruction).

B.2.1 Input from Current Students

Input from current students is obtained via several mechanisms. First, we have student representatives on key committees, including UGSC and the Curriculum Committee. Second, an open undergraduate forum is held each year to discuss all aspects of our programs, including its objectives. The forum is held in the spring quarter and is attended by interested students, key faculty members, and the professional undergraduate advisors. Announcements about the forum are made widely, especially on the electronic (student) newsgroups to ensure wide participation. Following the forum, a summary of the discussion is posted on the newsgroups by the chair of UGSC; this usually leads to further extended discussions where students (including those who could not attend the forum) further express their opinions and ideas on the program. Third, before graduation, students (in the CSE program) are required to complete an exit survey. The survey consists of two parallel sets of questions. The first asks the student to rate the importance (on a scale of 'very unimportant' to 'very important') of each of our objectives and outcomes; the second asks the student to rate how effective the program was, in his or her particular case, in meeting that objective or outcome.

B.2.2 Input from Alumni

Input from alumni is obtained by means of an anonymous (hard-copy) survey. Each year this survey is mailed out to alumni who graduated either two years ago or six years ago; this allows us to gather input from alumni who graduated relatively recently as well as some who graduated a while ago, without at the same time asking for input from the same group of people year after year.

The survey consists of two components, the first having to do with the respondent's overall educational experience at Ohio State, the second, which is the one that concerns us here, being similar to the exit survey, described above, for graduating students. Thus this component asks the respondent to rank the importance of each of our objectives and to rank how well, in his or her opinion, our department trained in the respondent with respect to the particular item.

B.2.3 Input from Employers

Employers (as well as potential employers, more on this in B.3) of our graduates are clearly one of most important constituent groups. But they also present some difficult challenges when it comes to seeking input. Perhaps the most important challenge is the question of privacy and related legal issues; i.e., typically employers are unwilling to provide answers to any questions involving the performance or preparation of employees. In addition, we face the problem of identifying the organizations that employ recent graduates of our programs.

We decided that alumni from our program who graduated 15 or more years ago are likely to be in supervisory positions and to be responsible for managing our more recent graduates. Hence each year we send out a survey to our alumni who graduated 15 years ago asking them to rank the importance of a number of general objectives/outcomes that would be appropriate, not just for computing programs, but for many science and engineering programs. For example, the survey includes questions on the importance of team-working skills, especially in multi-disciplinary settings. The survey does not include any questions about the abilities or preparation of any individuals the respondent may be currently supervising or may have supervised in the past; this alleviates the privacy and legal concerns.

We also obtain input in face-to-face discussions with members of our industrial advisory board. This group consists of senior managers from companies that are major employers of our graduates. The group meets at least once a year on campus and discusses all aspects of the department including research and teaching issues. As noted elsewhere in this report, both the original and recently revised set of objectives and outcomes was presented to and commented on by this group before approval by faculty.

In spite of these activities, we have felt that we need to do better with respect to getting input about our programs from employers of our graduates. In order to address this need, we have recently developed a new survey that on-campus recruiters of our students will be asked to complete. This is described in B.3.

B.2.4 Using the Assessments Results: Feedback Mechanisms

Good assessment instruments, by themselves, are not of much use. We also need to make use of the data they provide to identify and implement program improvements. The CIS Department has instituted a number of mechanisms to ensure this.

Each spring, the UGSC thoroughly discusses the data obtained during that academic year from the various survey instruments described above. Summary data from each of the survey instruments is posted on our web pages for students and faculty to comment upon. We have found that having, on these web pages, side-by-side displays of bar graphs representing the average values assigned by respondents to the importance of each of our outcomes versus how well the program prepared the respondents with respect that outcome, makes it particularly easy to identify problem areas. Specifically, any outcome that is assigned a high value with respect to importance and a low value with respect to preparation needs extra attention. Interestingly, the same displays also tell us which areas might do with *less* attention, i.e., the ones where the importance is rated low compared to the preparation. The table below shows an example of this data. This is a portion of the table that summarizes the results of the exit surveys completed by graduating CSE seniors for the year Au '99 – Su '00. (Since the data is for 1999-'00, the objectives and outcomes it refers to are the ones that were in effect at that time, i.e., the ones prior to the revision in Au '00.) The data corresponds to the objective related to providing graduates with an understanding of the human context in which computing activities take place, and the corresponding outcomes.

Objective/Outcome	**Objective Was Met**	**Objective Was Important**
	Disagree — Agree	Not Important — Important
3. To provide graduates with an understanding of the overall human context in which engineering and computing activities take place.	67% (N=67)	79% (N=65)
3a. Students will demonstrate an ability to communicate effectively.	70% (N=67)	93% (N=65)
3b. Students will obtain familiarity with basic ideas and contemporary issues in the social sciences and humanities.	69% (N=67)	61% (N=65)
3c. Students will obtain an understanding of social, professional and ethical issues related to computing.	66% (N=67)	72% (N=65)

The first column is the objective or outcome; the second averages the responses to the statement, "the objective/outcome was met in my individual case"; the third averages the responses to the statement, "the objective/outcome is important for me personally"; details of how the averaging was done are available on our web pages. The numbers in parentheses indicate how many seniors responded to that question. The scores for (3a) are striking: while the importance received 93%, the "objective was met" received only 70%.

But good education is, of course, much more than a matter of drawing bar graphs and identifying problems and solutions from them. The entire faculty must buy into the idea of continuous improvement of the whole program and be willing to participate in it. For this all-important purpose, we have implemented an innovative, and extremely effective process: The entire CIS curriculum is organized into a number of *course groups*; for example, all seven courses related to software engineering issues (including the beginning sequence discussed in Section A) form the software engineering course group; the four courses (CIS 681, 781, 782, 881) having to do with computer graphics topics form the graphics course group.

Once every two to three years, the faculty coordinators of the courses in a given group are responsible for producing a *Course Group Report* (CGR) and presenting it to the Curriculum Committee. The report is expected to address such questions as whether the courses are meeting their objectives (see discussion in B.1.3 and Table 1); whether prerequisites are appropriate; student reactions to the courses in the group based on SEIs, data from the surveys discussed above, etc.; relation to the rest of the program; and whether the courses in the group are helping meet the overall program objectives as intended. The CGR also provides information about the group's ideas for possible changes in the courses.

The CGR mechanism ensures that the right group of faculty is involved in discussions about each group of courses. And it also ensures that they take account of how the relevant data collected using the various assessment instruments, as well as how this group of courses fits in with the overall curriculum and overall program objectives. A number of CGRs have been produced, and are all available on our web pages. A quick look at any of them will demonstrate the care and dedication that faculty groups have put into producing these reports. These CGRs also serve to bring new faculty up to speed on how the particular courses the new faculty member might be involved with are related to other courses in the department.

B.3 Program Improvements

In this section we will briefly consider some of the improvements that have resulted from using the various assessment and feedback mechanisms described in B.2 and B.3.

B.3.1 Improvements in Program Objectives

Two items that are of particular concern to employers of our graduates, based on survey results, are communication skills and team-working skills. These items were included in the first version of our objectives and outcomes statement. However, they were mixed in with other somewhat unrelated outcomes. Given the importance of these outcomes to the ability of our graduates to perform well in their professional careers, we introduced a new high-level objective with the related outcomes being communication and team-working skills, into our objectives and outcomes statements. Indeed, this is the key difference between our original objectives statement and the current one.

B.3.2 Improvements in the Curriculum

Given the greater importance that our objectives and outcomes statement attaches to communication skills as well as the feedback we have received on this from our assessment instruments (see B.2.4), we recently revised CIS 601 was recently modified to include *formal debates* (in addition to individual presentations by students). This change, we believe, will enable our students to become more adept at thinking, *and communicating* on their feet.

Another important feedback we have received from the surveys, as well as from the undergraduate forum, is the importance of, and the need for courses that deal with, new technologies. This is especially important in a fast moving field such as computing. Partly in response to this, CIS 694V, a course on multimedia networking was developed. Given the enormous potential for practical applications using multi-media networking, many of graduates are likely to be interested in this field.

Because of space limitations, we will not discuss other changes in the curriculum made in response to the feedback from our constituents.

B.3.3 Improvements in Process

We already noted (B.2.3) the difficulties in obtaining feedback from employers. Recently, we have developed a new survey instrument that we believe will help with this problem. This survey is directed toward on-campus recruiters. Like our other surveys, this survey will ask the respondent to rank the importance, with respect to the needs of their particular organization, of our various objectives and outcomes. We expect to receive good data from this instrument because privacy issues should not be a serious concern since the respondents are recruiters, not current employers of our graduates; moreover, return rates should be relatively high since recruiters would probably feel it is in their interest to help us improve our program.

C. Other Criteria

In this section we briefly consider each of the criteria listed in the call for nomination and explain how the CIS Department fares with respect to each.

Most of the criteria listed under 'Commitment to excellence in teaching and learning' have been addressed in Sections A and B. The ones that were not addressed are:

Diversity: Computer Science suffers from a nationwide underrepresentation by women and ethnic minorities. To proactively address this problem, the CIS Department applied for and obtained a competetive grant from the GE Foundation. This grant, originally for three years and successfully renewed this past year, provides support to attract Undergrad and graduate students from underrepresented groups to participate in research. The CIS Department cost-shares this grant extensively and is committed to continuing certain aspects of this program beyond the grant period.

Budgetary resources: The demand for CIS and CSE majors has exploded in the last few years and our budget has not risen in response with the demand. For this reason, we have been forced to limit entrace into our courses, and deny to many students admission to courses in our department and to the various majors for which our department is responsible. In spite of this, we have increased credit-hour production by 35% since FY'95 and increased the number of graduates of our BS programs by 30% over the same period. Thus, measured in terms of dollars, the CIS department's (or perhaps more accurately, the university's) efforts in investing adequate budgetary resources to meet student demand should probably be rated 'fair'. Nevertheless, as explained in the earlier sections, in terms of ability to work innovatively with a limited budget, our effort should be rated 'excellent'.

All the criteria listed under 'Self-evaluation of teaching and learning quality' have been addressed in Section B. The ones listed under 'Development of faculty teaching' are:

Support for new faculty and TAs: New GTAs attend a week-long orientation at the start of fall quarter; this is similar to the GTA-orientation in many departments. With respect to new faculty, one point worth reiterating is the Course Group Report mechanism discussed in Section B.2.4. Since all the CGRs are easily avialable on our web pages, the new faculty member can peruse the ones for the groups that his or her course(s) fit into; this gives the new faculty member a good head start on how to approach the courses he or she will be teaching. New faculty are observed informally by the department chair during a lecture they give to new graduate students in the Fall. New TAs are observed by their course coordinators. The chair also specifically encourages professional development related to teaching, and has supported faculty attendance at several teaching related conferences and workshops.

Utilizing university resources: The department participates in, and often takes the lead in, discussions among various departments of the College of Engineering in developing new assessment instruments, etc. Indeed, the college has expressed interest in using some of the summary mechanisms developed in our department (and described in B.2.4).

Teaching technique research etc.: The most noteworthy example of this are the activities relating to our beginning sequence, as detailed in Section A.

Criteria listed under 'Resources for students':

Access to teachers and advisors: All our students, faculty, and advisors rely heavily upon electronic communication. Every class has an active electronic newsgroup to encourage discussions among students in the class and the instructor. Often students who took the course in past quarters jump into the discussions (the newsgroups are open to all members of the department) with their own insights. With respect to electronic interaction among students and faculty and advisors, CIS would easily rank among the best in the university.

Support services: We have a undergraduate advising office staffed by two full-time professional advisors and a GAA. They provide prompt guidance to students concerning support services as needed. Our lead advisor is very active in national-level advising organizations. She is very much in touch with, and a contributor to, the development of, best practices in advising.

Learning opportunities outside the classroom: High ability students are encouraged to attend research group meetings involving topics of interest to them. This often leads to undergrad research in the students' junior or senior years. The Department also has student chapters of computer science professional organizations that offer opportunities to hone skills and attract outside visitors to enrich their educational experience.

Keeping students informed: We make very effective use of our web pages to keep students informed of all issues involving their educational programs. Agendas and minutes and of key committee meetings (Curriculum Committee, Computer Committee, UGSC) are all posted on student newsgroups.

D. Summary

While the CIS Department cannot, and we do not in this statement, claim to excel in each of the criteria listed in the call for nominations to the UDTEA, we strongly believe that we score very highly in several of the categories. In particular, we believe that in the criteria having to do with excellence and innovation in teaching and learning, as well as in the criteria to do with self-evaluation and improvement of program quality, our achievements are truly superior. We hope the Selection Committee agrees with this.