


# Freedom In Teaching Applied To Computer Science Education And Its Relationship To Accreditation And Diversity

Osama Shata, (E-mail: sosama@qu.edu.qa), University of Qatar, Qatar

## ABSTRACT

*Much of the literature reviewed on the subject of freedom in teaching seems to emphasize that it is not only highly desirable and appreciated but deeply embedded in the core of the teaching profession. Although freedom in teaching has been used to mean freedom of academic institutions, instructors and students, but it has focused mostly on freedom of instructors in classrooms to discuss materials relevant to their courses and in their research. The intention behind this paper is not to look for evidence to support what is already known, but rather to contribute to the understanding of the subject by extending what is meant by the term freedom in teaching to cover freedom of disciplines. This paper seeks to use this extension to propose that freedom in teaching can help addressing and responding to many challenges that face a rapidly changing discipline such as the discipline of Computer Science. The paper focuses on how freedom in teaching computer science at both program and course levels may play a pivot role in responding to some of the discipline's challenges. The paper also seeks to link freedom in teaching to issues such as diversity, accreditation and learning objects. The paper concludes by discussing the disadvantages and burdens that may come with freedom in teaching. Although that this paper focuses on the discipline of Computer Science as a case to study, but the arguments and discussion may be generalized to cover other disciplines that face similar challenges.*

## INTRODUCTION

 Computer Science is usually characterized as a highly active and rapidly changing discipline. However, the discipline faces many challenges. Until recently, computer science departments used to report continuous increase in numbers of undergraduates applying and joining their programs. Now, many departments are concerned about the sharp decline of computer science majors.

Many universities and departments are stepping up efforts to promote computer science (CS). Research is conducted to identify problem areas and how to respond to them. Such efforts and research focus, in many cases, on areas of: gender imbalance, interest and motivation, culture differences, theoretical and mathematical content, teaching and learning styles, and adjustments in programs and courses. We believe that all those areas may be covered under one broad header: recognizing the diversity of computer science. Recognizing diversity and addressing problems areas goes fast if is supported by freedom to: think, investigate, analyze, argue, propose, design, practice, apply, and invent (outcomes of good teaching).

Hence, we believe that freedom in teaching provides the foundation for supporting this diversity through flexible curricula, courses, and teaching practices. However, this freedom is restricted by the traditional definition of the discipline which, in our opinion, does not provide enough support to the diversity of CS, and which needs to be re-defined to allow freedom in teaching be applied to the discipline itself.

This paper begins with discussing possible definitions for “freedom in teaching computer science (FTCS)” and uses this discussion as a frame to introduce faces of this freedom at both program and course levels, such as requirements of accreditation criteria and developing learning objects. The paper emphasizes that freedom in teaching

at both program and course levels plays a critical role in responding to some of the problem areas and challenges to the CS discipline.

This paper also discusses the burden that freedom in teaching may bring, especially to individuals. Although freedom in teaching is a general requirement, but has its responsibilities since that much of current thinking in college education suggests that freedom in teaching goes inline with innovation in teaching. The matter that may lead to unrealistic expectations to be anticipated from individuals who practice freedom in teaching, and hence adds considerable burden on them.

Next we introduce possible definitions for “freedom in teaching”.

### **WHAT IS FREEDOM IN TEACHING – APPLIED TO COMPUTER SCIENCE?**

Freedom is such an abstract word that has various meanings, scopes, and dimensions to individuals depending on each individual’s background, psychology, believes, and ethics. However, most individuals shall relate its meaning to the condition of acting without restraints.

Teaching computer science is a process that involves introducing material on computer science by professors and instructors (teachers) to students and which takes place in academic institutions. Hence, freedom in teaching computer science may have various interpretations by the main stakeholders: institutions, professors, and students.

Academic institutions in general, including those that offer computer science programs, assume that they bear a heavy responsibility to maintain and defend the human thoughts and conceptions, to achieve rapid and continuous progress in finding and comprehending new knowledge and technology, and to teach students to assist and contribute positively to the society. Institutions, in general, interpret this assumption and responsibility as their rights to determine, on academic and scientific basis, what may be taught, and how it shall be taught in a way that preserves, renews and strengthen their distinctive identity. Institutions must be responsible for a considerable degree of self-determining what may be taught in the computer science programs. The programs must take into consideration: cultural, economic, or ideological factors.

Professors may have their own view of freedom in teaching computer science. Some may link it to academic freedom, and their rights to teach, pursue, and discuss knowledge without restriction or interference, as by the institutions which employ them. However, most of them understand that they are responsible for teaching courses (elements of the academic programs as set and approved by institutions). Hence, freedom in teaching is not applicable to every activity, and without accountability it can be fruitless, useless, and unproductive. In this sense, professors must be responsible for a considerable degree of self-determining what may be taught in the courses to serve the program goals and be consistent with the purposes of the institutions. Others, may link freedom in teaching computer science to the absence of constraints in selecting the appropriate teaching pedagogy, use of technology and what technology to incorporate, assessment, etc.

Some students may interpret freedom in teaching computer science as the use of technology to make their tasks easier. For example, the use of computer conferencing, on-line tutorials and assessment to give them freedom to learn at the time they like and according to their pace. In addition, some students may link freedom of teaching computer science to the ability of professors to recognize diversity of students (e.g. to offer them various learning materials and assessment methods to select from). Other students may interpret freedom in teaching as the assurance that they may express themselves freely in the classroom without having to worry that their evaluation or achievement toward a degree will be affected due to a disagreement or conflict of opinions, believes or thoughts with their instructors.

Unfortunately it is difficult to reach one definition by consensus for freedom in teaching computer science.

In this paper we will not argue one definition for FTCS, but rather we will introduce what we may consider some forms of applying freedom in teaching at program and course levels since this may be of interest to all stakeholders, and discuss the relationship between those forms and some of the problem areas / concerns to computer science programs. The discussion will cover freedom in teaching at program and course levels, linking it to the issues of accreditation and learning objects.

Next, some of the identified concerns to computer science programs will be presented.

### **SOME IDENTIFIED CONCERNS TO COMPUTER SCIENCE PROGRAMS**

The first concern we list here is the sharp decline in numbers of students joining computer science majors. Vegso [17], in his article on declining interest in computer science among freshmen, states that: “An analysis of survey results from the Higher Education Research Institute at the University of California at Los Angeles (HERI/UCLA) indicates that the popularity of computer science (CS) as a major among incoming freshmen at all undergraduate institutions has dropped significantly in the past four years”. Foster [11] claims that: “New data show students' interest in the discipline is in a free fall”. This sharp decline in number of students majoring in computer science may be due to the considerable amount of theory involved, the large number of mathematics courses required, and to the competition the discipline is experiencing from other more appealing disciplines such as information technology, information science, and information systems which are attractive, less challenging in mathematics, and more applied alternatives to computer science [11].

The second concern we list here is the wide and continuously increasing gap between the numbers of men and women joining computer science. According to Vegso [17]: “Overall, interest in CS among women fell 80 percent between 1998 and 2004, and 93 percent since its peak in 1982.” It seems that the general drop in students joining the discipline of computer science have made the discipline unappealing to female students in unseen level. In addition, in their study, Fisher, Margolis and Miller [10] suggest that women students, in contrast to men, seem to be discouraged from computers as programming devices, and prefer to focus on the social aspects of computing and its applications.

The last concern we list here is the difference in culture: a class may be composed of students that are from different cultures who may differ in their learning styles needs. Research points out that there is a link between culture and learning style. Slay [14] claims that “research indicate that Chinese students would find it easier to understand and apply theoretical principles within programming than would a similar group of Western students. Chin & Chang [5] point out that Asian students differ from Anglo-Saxon students when it comes to web-based learning, “More Asian students are attracted to Web-based learning because it is an innovative idea for facilitating learning”.

The conclusion here is that computer science is a discipline with various aspects of diversity such as interest, gender, and cultural. A pivot point in responding to diversity is freedom.

We discuss next what may be considered examples of steps taken towards applying freedom in teaching computer science at both program and course levels, and which may have helped in addressing the above concerns.

### **THE RELATIONSHIP BETWEEN FREEDOM IN TEACHING COMPUTER SCIENCE AND THOSE CONCERNS**

As we noted earlier that freedom is a pivot point in addressing diversity, we believe that scientists and researchers have applied freedom in teaching to the discipline of computer science in various ways. The matter that may have helped in addressing the above concerns. Researchers may have not explicitly used the term “freedom in teaching computer science” in their research and applications, however, we find that many of recent research and modern perspectives to teaching computer science are actually results and faces of applying FTCS, at both program and course levels. A main goal of this paper is to capture, through different examples and applications, the link between freedom in teaching and responding to diversity.

We introduce next examples on what may be considered steps taken towards applying freedom in teaching computer science at both program and course levels.

### **The Program Level**

There is continuous research being conducted by the computer science academic community, on curriculum and teaching methodologies, to respond to the “evolutionary” nature of the discipline.

We find that one of the important steps taken by the computer science academic community to respond to this evolutionary change in the discipline is the more flexibility (freedom) offered in the accreditation criteria. Some computer science accreditation bodies have put emphasis on individual learning, the technical aspects as well as human aspect of the use of computers in the wider world where computers are used, and recognizing diversity of learners. We consider this as applying freedom in teaching to computer science. We believe that this freedom may have been recognized for the computer science discipline more than in other disciplines. For example, we find that appreciated effort has been made to the current Computer Science Criteria in the ABET accreditation process and operational philosophy [1] (ABET is the body responsible for accrediting all undergraduate Engineering, Engineering Technology, Applied Science and Computer Science Programs in the US). In designing this criteria, ABET has striven to keep the criteria flexible in order to adapt to new forms of computer science education or interaction with other sciences and disciplines. The criteria has not set a predefined set of specific program’s outcomes, but rather has left it to each program to define its objectives and outcomes (and courses within a flexible frame that consists of three primary components: required courses in the major, math and science courses, and breadth in the liberal arts, in addition to free elective). In comparison with other engineering disciplines, there is a set of specific outcomes that engineering programs must demonstrate. The ABET accreditation criteria for engineering programs, Criterion 3: Program Outcomes and Assessment lists 11 fixed program outcomes for accredited programs to have their graduates to demonstrate. We find this an application of teaching in computer science at program level to reflect a better pedagogical model, where students are given greater freedom and greater responsibility for their own learning. This flexibility (freedom) in designing CS programs, and which is supported by the accreditation criteria, can help addressing the problem of diverse interests since it allows providing various areas within the curriculum to suit various students’ needs and interests.

A second example is the many research focusing on attracting and retaining women in computer science programs [2, 3, 4, 12]. Such research suggests techniques, tips, and approaches on how to make computer science courses more appealing to women. Again, we find this a call to freedom in teaching computer science, where teachers may be free to use various teaching models (e.g. lab work, collaborative tasks, etc.). The matter that can help addressing the problem of the wide and continuously increasing gap between the numbers of men and women joining computer science.

### **The Course Level**

Courses are elements of programs, and the freedom in teaching computer science at the program level, as discussed in 4.1. Above, is reflected on teaching computer science courses.

For example, there is a growing tendency away from a traditional teacher-centered perspective in higher education, towards a modern student-centered perspective. The teacher-centered perspective focuses on the objectives of a course and looks at what the teacher (one individual) does and wants to cover in the course. Whereas the student-centered perspective focuses the learning outcomes of the course and on what the students (many individuals) will be able to do. This modern perspective requires the teacher to take the students’ perspective and make a realistic estimate of what students are expected to know and be able to do by the end of the course (i.e. taking diversity of students into consideration). But does not this modern perspective actually mean freedom in teaching (e.g. selecting content and various and appropriate assessment measures to suit diverse students) to respond to the problems of diverse interests and cultures?

A second example we introduce here to emphasize the recognition of freedom in teaching computer science at the course level is the research and development of learning objects. One of the techniques used by computer science academic to improve the effectiveness of teaching and learning is by sharing peer reviewed online learning materials. Developing an online course is expensive and is time consuming. Some courses of the same institutions may have some parts in common, and corresponding courses in different institution also have some common parts in common. So it is practical that those common parts be developed only once. Downes [9] clarifies that “Although courses may share elements in common, it is rare to find two courses from two institutions that share the same, and only the same, set of elements. Thus, courses themselves are not suitable candidates for sharing”. Hence, the units to be shared must be smaller than the complete courses (e.g. text dealing with a specific topic, objectives / outcomes, assessment questions on that text, audio file (if any), video clip (if any)). Those units are named, using the current terminology of the field, “learning objects”. We consider such applications on the use of learning objects, at levels smaller than courses to recognize diversity of students and courses, a form of freedom in teaching computer science.

A final example we list here on freedom in teaching computer science is the research on “Learning Styles”, although this applies to other disciplines as well, where one of its primary goals is to understand the link between culture and learning to help choosing instructional techniques and curricula to meet the need of all students. A teacher’s critical goal should be to discover the specific learning needs of his or her individual students. Again we find this evidence on the call and applications of freedom in teaching which contributes to responding to the problem of differences / diverse of cultures.

The conclusion here is that freedom in teaching computer science has been realized, recognized and applied by scientists and researchers. This freedom is a must to allow cultural, gender, and interest preferences within the teaching of computer science, and hence help addressing some of the concerns of the discipline.

We find that the above steps towards applying freedom in teaching CS were limited and restricted by the traditional definition of the discipline and which does not provide enough support to diversity in CS. Hence, we argue next the need to apply freedom in teaching to the definition of the discipline and to break some of the constraints that are imposed by the traditional definition.

## **FREEDOM IN TEACHING AND THE TRADITIONAL DEFINITION OF THE DISCIPLINE OF COMPUTER SCIENCE**

Many research papers and articles on computer science begin with sentences indicating that the discipline is changing in unprecedented way, and that computer science is a highly active and fast changing discipline that is facing many challenges. We listed earlier some of those challenges and introduced examples that we considered steps and applications of FTCS and which help addressing those challenges. However, we believe that such steps and applications are not enough to solve the troubles facing the discipline since many universities are still reporting sharp decline in numbers of undergraduates joining computer science programs.

We claim that freedom in teaching computer science has been always limited because it is constrained and restricted by the traditional definition of the discipline when it was taught as part of mathematics departments. Many computer science departments, when introducing or defining the discipline in their catalogues or web sites, highlight its traditional roots in mathematics, logic, and algorithms, and focus on its theoretical aspects [7, 8]. When applying FTCS to programs and courses, this freedom is still constrained by this traditional view. As a result, such programs and courses are not appealing to students. Even some accreditation criteria are also constrained by this traditional definition. For example, the ABET has striven to keep the computer science accreditation criteria flexible, however, it was still affected by the traditional definition of the discipline when it required 15-18 semester hours of mathematics. One can grasp that computer science undergraduate’s need knowledge in areas such as discrete mathematics, probability, statistics, differential and integral calculus. But why specifying a minimum number of credit hours in mathematics? What if students can cover the needed knowledge in the above areas in fewer hours? A second example we list here is the insistence of some programs to keep teaching courses such as compiler design and formal methods, and dedicate effort to justify this. In the early days of computers there was much work to be done in those areas. Do we still need to teach them on a wide scale nowadays?

We believe that we need a new definition for the discipline to highlight its: applications in processing information, human factors, response to technological problems of the world in other disciplines, links to humanities, educations, law, and other disciplines. Programs then can teach courses in those areas with the existence of streams that may teach more abstract areas such as mathematics comprehensively for those interested and as needed by the job market. The need for this new definition may be appreciated by the fact that many of computer science graduates work as end users, applications developers, troubleshooters. In contrast, how many of them work in designing new compilers, or new programming languages? In addition, to the some, there is no sharp difference between computer science and information technology. In many cases both terms are used interchangeably. Is it possible to consider a definition that combines both or emphasizes the relationship between them? Actually, to be more accurate, we need to formalize a generic and flexible definition that would express a continuously changing, heterogeneous, and many-sided discipline and which attracts individuals with deviating, hybrid, and mixed interests. Computer science is a relatively new discipline compared to other disciplines such physics, chemistry, or biology. There is still much debate about it, and the some are challenging considering it as a “science” since it does not follow the established scientific procedures. According to Wikipedia [18]: “Scientific method or procedure is considered fundamental to scientific investigation and the acquisition of new knowledge based upon physical evidence. Scientists use observations, hypotheses and logic to propose explanations for natural phenomena in the form of theories. Predictions from these theories that can be reproducibly tested by experiment are the basis for developing new technology”. So the some argue that computer science is more like the discipline of mathematics. Probably that is why the some prefer not to give a detailed and solid definition to computer science and rather prefer to list areas in which computer science is concerned with such as algorithms, databases, artificial intelligence, data structures. For example, the web site of The Department of Computer Science at the University of Maryland [16] shows: “What is Computer Science? Computer science is the study of computers and computational systems: their theory, design, development, and application. Principal areas within computer science include artificial intelligence, computer systems and networks, database systems, human factors, networks, numerical analysis, programming languages, software engineering, and theory of computing” . We do not intent to argue for or against considering computer science as a science or not. However, we find this argument and the lack of a solid definition for the discipline an opportunity for academics to discuss a generic definition that suits various purposes and recognizes various interests. A starting point could be to realize that computer science is surviving because of its wide applications and its high synergy with many other disciplines. The emphasis should be on its applied aspect. The fact that computer science has its roots in mathematics does not mean that the emphasis should be on its theoretical face. One can argue that computer science also has its roots in engineering which is the application of science. In addition, computer science is not only about studying hardware and software to produce good programmers. Many useful applications, some of them are even distributed as freeware and shareware, have been developed by people who know only one programming language and know more about other areas than what they know about computer science. May be it is time to look to computer science as a service discipline to other disciplines in the same way that service courses, in academic institutions, are required by all programs. Students take service courses such as English as language and mathematics to help students to acquire some academic skills they need in order to operate successfully within their major. Part of the popularity of computer science is that it represents a popular technology that is needed by almost all other disciplines at various levels. Many students from various majors prefer to have their minor in CS. This is not a call to eliminate mathematics and theory courses from computer science programs, but rather this is a suggestion to emphasize the practical face of the discipline. It is a special discipline where one end is its theory and the other end is its practical implementations and applications in many other disciplines. Along that way we can find some points that are closer to theory, others in between, others are closer to implementations and applications. Those points can be represented and implemented by various streams to better serve various interests, gender, and cultures (i.e. CS diversity) and the job market. We may end up with a more realistic and flexible definition to the discipline that allows for more freedom in designing and teaching programs and courses. In which case, we believe that many of the troubles facing the discipline may be solved.

## **THE BURDEN OF FREEDOM IN TEACHING**

When freedom in teaching is viewed from the perspective discussed throughout this paper, it is considered a positive phenomenon and a productive process that follows an evolutionary path. While we believe that it is, we also find that it has many burdens correlative with it.

This section is an initial attempt to make sense of the burdens of freedom in teaching as it is introduced in this paper.

When a department is given freedom to design its computer Science program and to set its learning outcomes, the program administration will establish objectives and learning outcomes. The program administration has the burden of justifying the effectiveness of its program and the appropriateness of the objectives and learning outcomes to many parties, for example:

- to the institution administration, and that the program objectives derive from and support the purpose of the institution,
- to the program constituents, and that the learning outcomes would mean that graduates of the program will be able to function successfully in different work places with diverse and complex environments, to deal professionally with state of the art technology, to reason critically, and to communicate efficiently.
- to students, that the program's curriculum is carefully designed to provide them with the necessary knowledge and skills, and to meet diverse of needs.

This burden is much more realized in the case of CS programs when compared to other programs that enjoy less freedom. This perception draws on my own experience as a professor in a Department of Computer Science and Engineering that offers two degrees: Bachelor of Computer Science (CS) and Bachelor of Computer Engineering (CE). From personal experience, administration of the CE program adopted the set of learning outcomes as defined and listed by the ABET, and hence did not have to suffer extended arguments with administration, constituents and students. On the other hand, the administration of the CS program conducted lengthy and frequent meetings with interested parties to justify the set of learning outcomes that it has created and adopted.

In addition, the evolutionary nature of the discipline of computer science and the continuous rapid innovations in computer hardware and software supported the view that good teaching computer science should also be innovative, and more freedom in teaching CS, in the way explained in this paper, would definitely mean more innovation in teaching CS. Thus, we find that much of the current thinking in CS college education considers freedom in teaching CS is synonymous with (or a major reason for) innovation in teaching CS. In the absence of a clear definition of innovation in teaching, many adopt dictionary definitions such as that of The Free Dictionary [15] and which states that innovation is “The act of introducing something new” and “Something newly introduced”. There are continuous increasing demands from administrations of institutions asking instructors for innovation in teaching (i.e. using new teaching methods) to achieve greater productivity with better quality. In many cases the administration itself has unclear expectations or interpretation for these demands and it is left to the instructor, as part of the freedom in teaching, to interpret these demands and respond to them in order to be assessed accordingly. This is a very complex burden for instructors to deal with or to implement. How to measure good teaching productivity or quality? Would depending on number of students successfully completing a course be a proper measuring scheme? (taking in consideration that instructors can easily lower their assessment standards). How about handling classes with large enrollments? Or dealing with larger number of students with disabilities?

Furthermore, what is meant by “new teaching methods”? Does it mean using technology in classrooms? We can find that what one instructor may consider a new teaching method is considered old by another. Would using slides or laptop presentations instead of a blackboard be considered a new teaching method. How about depending on on-line resources rather than few printed ones? Would this be considered a new teaching method? Hughes [13] states that: “Many teachers will have been introduced to ‘new’ teaching methods in conferences and staff development sessions only to discover that they have been using that method, which they may feel they have personally developed, for some considerable time. This can produce a range of reactions in the individual, from feeling that they are not

learning anything new, to feeling cheated that they had not been recognized, in the past, as innovators. This, in turn, may have an effect on their own level of motivation, adversely influencing the quality of their teaching practice”.

We find that it is unrealistic to associate freedom in teaching with the literate / traditional meaning of innovation which denotes new methods, and assess instructors’ teaching performance accordingly.

We claim that this type of responsibility that came with freedom in teaching has generated heavy burdens on instructors. We have realized and practiced that this type of “more freedom in teaching” has caused many instructors to be more exhausted and depressed. This subjective claim is based on personal experience rather than solid statistical sampling and examination.

We believe that a realistic view of innovation in teaching should not assume a continuous increase of some kind in the activities carried out by instructors. We find that innovation in teaching as a result of freedom in teaching should include any improvement in teaching practices. Improving a set of examples on some subject matter so that it becomes clearer and easier for students may be considered innovation. Actually, from our experience and from discussions with colleagues, we support the view that many of the improvements in the teaching process resulted from the experiences instructors accumulated from teaching the same set of courses repeatedly. This gives instructors the opportunity to re-consider their activities and discover opportunities for improving them. Actually, this view goes inline with a comprehensive definition for innovation and which may be found in the Wikipedia [18]: “Innovation is defined in the dictionary as *the process of making changes to something established by introducing something new*. The term Innovation refers to both radical and incremental changes to products, processes or services. Innovation is often confused with the term Invent which is defined in the dictionary as creating something new, something that has never existed before. Innovation is supposed to add value, but it can also be positive or negative. Most ideas generated through the innovation process are destroyed and organizations that do not innovate effectively are destroyed by those that do. Positive innovation can be defined as - *the process of making changes to something established by introducing something new ...*”

Assessing innovation in teaching, based on freedom in teaching, should be according to some professional standards set by professional authorities. For example, a realistic and acceptable measure for innovation in teaching is coping with accreditation requirements. Each course has a defined list of learning outcomes that can be assessed. When the course assessment shows that students passed the course are able to demonstrate the competencies defined in the course’s list of learning outcomes in an acceptable level, or noticeable progress in students’ competencies, then this may be accepted as innovation in teaching.

Another source of burdens and responsibilities that come with freedom in teaching is meeting student’s expectations. Students may have their interpretation of freedom in teaching (e.g. using state of the art technology, special technology products, etc.) and set their expectations from a course accordingly. If some of the students’ expectations do not match with the realities of the course then the instructors have the burden of changing student expectations and the consequences of these.

Other, but less significant, causes of burdens of freedom in teaching may include: Administration may have the view that freedom in teaching enables instructors to manage their courses better in less time, so increasing their workload is justifiable.

Also, some people relate more freedom to the desire of some individuals to use it to remain disorder and irresponsible, hence those who enjoy freedom in teaching must be carefully assessed and supervised. This places more burden on faculty who must report regular improvements in their teaching assessments However, we believe that administration in academic institutions do not widely adopt this view, but rather trust their instructors.



## CONCLUSION

Traditionally, the term “freedom in teaching” has been used to mean the academic freedom of institutions, instructors, and students. But the term is used increasingly to mean instructor’s freedom. This freedom is recognized and appreciated to create a healthy teaching environment. This paper has considered the freedom of disciplines. Rapidly changing disciplines face many challenges. This paper has focused on the discipline of computer science, as a case, which face severe challenges such as continuous reduction in enrollments, gap between male and female students, and coping with diversity. Much research has been conducted to address those challenges without considerable improvement. This paper has argued that most proposed solutions were actually a form of freedom in teaching applied to the discipline itself. Hence, the paper aimed to provide a discussion of some of those solutions and relate them to freeing the discipline from traditional constraints, and used this discussion to stress that freedom in teaching should be broadened to cover the disciplines (in addition to institution, instructors, and students). The paper also argued that freedom in teaching applied to disciplines may be constrained by their traditional definitions. Hence, changing the definitions themselves may be necessary. The paper was concluded by presenting some of the burdens and responsibilities that come with freedom in teaching and proposed solutions to some of them.

## REFERENCES

1. ABET. Online resource retrieved on March 19, 2006 from <http://www.abet.org/>
2. Belsie, L. (1998). Computers for the rest of us: Girls want to have fun on the computer too. *Christian Science Monitor*.
3. Borg, A. (1998). What draws women to and keeps women in computing? In C. C. Selby (Ed.) *Women in science and engineering: Choices for success*. New York: *New York Academy of Sciences*..
4. Brown, J., Andreae, P., Biddle, R., and Tempero, E. (1997). Women in introductory computer science: Experience at Victoria University of Wellington. *Proceedings of the ACM SIGSCE Conference*.
5. Chin, K. L. & Chang, V. (1999). The Impact of Cultural Differences on Web-based Learning. *Proceedings of the 2nd Western Australian Workshop on Information Systems Research*. Online resource retrieved on April 1, 2006 from <http://wawisr01.uwa.edu.au/1999/ChinChang.pdf>
6. Clarke, V. (1992). Strategies for involving girls in computer science. In C. D. Martin and E. Murchie-Beyma, (Eds.) *In search of gender-free paradigms for computer science education*. Eugene, OR: *International Society for Technology in Education*, pp. 71-86.
7. Colgate Computer Science Department web site. Online resource retrieved on March 20, 2006 from <http://cs.colgate.edu/>
8. Department of Computer Science at Washington and Lee University. Online resource retrieved on March 21, 2006 from <http://cs.wlu.edu/>
9. Downes, Stephen (2000). Learning Objects. Online resource retrieved on March 22, 2006 from [http://www.atl.ualberta.ca/downes/naweb/Learning\\_Objects.htm](http://www.atl.ualberta.ca/downes/naweb/Learning_Objects.htm)
10. Fisher, A., Margolis, J., & Miller, F. (1997). Undergraduate women in computer science: Experience, motivation, and culture. *Proceedings of the twenty-eighth SIGCSE technical symposium on computer science education*, pp. 106-110. Online resource retrieved on March 23, 2006 from <http://www.acm.org/pubs/contents/proceedings/cse/268084/>
11. Foster, Andera L. (2005). “Student Interest in Computer Science Plummet -Technology companies struggle to fill vacant positions” in the *Chronicle of Higher Education*, Section: Information Technology, Volume 51, Issue 38, Page A31 Online resource retrieved on March 20, 2006 from <http://chronicle.com/free/v51/i38/38a03101.htm>
12. Haller, S. M., & Fossum, T. V. (1998). Retaining women in CS with accessible role models. *Proceedings of the ACM SIGSCE Conference*.
13. Hughes, Susan (2003). What is innovation in teaching. Online resource retrieved on April 3, 2006 from <http://www.business.heacademy.ac.uk/resources/reflect/conf/2003/hughes>
14. Slay, J. (2000). Implementing modern approaches to teaching computer science: a life-long learning perspective. 16th World Computer Congress, Beijing, August 21-25.
15. The Free Dictionary. Online resource available at: <http://www.thefreedictionary.com/innovation>

16. University of Maryland – The Computer Science Department web site. Online resource retrieved on March 25, 2006 from <http://www.cs.umd.edu/Ugrad/OldIntro/>
17. Vegso, Jay (2005). Interest in CS as a Major Drops Among Incoming Freshmen in *Computing Research News*, Vol. 17/No.
18. Wikipedia, the free encyclopedia. Online resource retrieved on April 3, 2006 from: [http://en.wikipedia.org/wiki/Scientific\\_method](http://en.wikipedia.org/wiki/Scientific_method)

**NOTES**