PROPOSAL FOR NEW INSTRUCTIONAL PROGRAM
submitted by

UNIVERSITY OF MARYLAND AT COLLEGE PARK, MARYLAND

COLLEGE OF COMPUTER, MATHEMATICAL AND NATURAL SCIENCES
and
COLLEGE OF EDUCATION

Computer Science Secondary Education Teacher Preparation Programs

Currently there is no undergraduate degree program that enables one to become certified to teach computer science in Maryland, a gap that contributes to the state's pressing STEM demands within our economy. High schools have one of two choices: not offer Computer Science or have a teacher from another discipline teach computer science. Often high schools will task mathematics, business or technology teachers to cover computer science needs, but this gives inconsistent treatment from school to school across the state. Moreover, teachers with little or no experience in the field can hardly give their high school students realistic guidance on how to prepare for entry to this critical STEM area. To remedy this situation, the College of Education and College of Computer, Mathematical and Natural Sciences (CMNS) jointly propose a coordinated suite of programs that provide the greatest number of pathways to prepare our students for teaching careers in computer science: Computer Science Education, Computer Science Education/Secondary Education, five year integrated program (IMCP), and 5th year program (MCert). By design and intent, all of these paths lead to teacher certification in computer science for grades 7-12. In this proposal, we concentrate on the two undergraduate pathways.

In order to be certified in a secondary education discipline, students must obtain a content-based degree. Therefore the core component of this proposal is introduction of a new undergraduate track within the Computer Science Department. While a science degree, it is specifically crafted to meet the objectives of Secondary Education as well, and thus is jointly advised and informed by College of Education partners. They, in turn, will take primary responsibility for administration of the second major component of this proposal, the Double Major in Secondary Education, for those CS Ed majors who further satisfy the proposed requirements for certification in this field. This model is comparable to that found in other disciplines in CMNS (such as Physics and Mathematics) that presently have education tracks.

PATHWAYS

Bachelor Degree in Computer Science – Traditional Track

The present computer science bachelor's degree requirements will remain unchanged, and will be referred to here as the Traditional Track. With the addition of an Education Track (to be described presently), all majors will have essentially identical curricular goals for the first two years, at which point plans must become nuanced for either the traditional or educational track. No special application is required for a CS major to participate in either the Traditional or Education track, though coordination with advisors is fully expected in order to ensure student plans meet benchmarks.

Bachelor Degree in Computer Science Education - Double Major in Secondary Education

While the CS Education Track could in some cases result in a stand-alone degree without the Double Major in Secondary Education, the expectation is that students who are intent on this track will actively pursue a Double Major in Secondary Education, which would make them eligible for state teaching certification. CS majors considering the Education Track will thus make application to (and receive advising from) the College of Education prior to registration for the first courses in that college, according to routine procedures that would be prescribed.

CS majors who are not accepted by the College of Education will not be offered the Education Track in the CS department; they will remain in the Traditional Track and be expected to meet all those requirements accordingly. Majors accepted to the Education Track who do not satisfy its course requirements may nevertheless graduate with a Traditional Track degree by meeting those requirements. It is possible that a major who has been accepted into the Education Track might satisfy its course requirements but not meet minimum graduation requirements for other reasons (say, concerning GPA) set by the College of Education; in this case the student would receive just the Bachelor Degree in Computer Science Education Track (not the Double Major.)

MCERT (Masters Certification) in Curriculum and Instruction of Computer Science

The College of Education already provides MCERT opportunities for teachers in the field who wish to return and expand their credentials in other areas. These are full year (13 month) commitments. The present proposal would add an MCERT specialization in Computer Science Education, consistent with all the requirements defined on other degree paths. The College of Education would take the lead in administering this program, and would rely upon the Computer Science Department for content delivery in the key courses as will be identified.

Five Year Integrated Bachelor/Masters Program

Undergraduate CS majors who anticipate that they may be interested in immediately obtaining advanced credentials may pursue the Five Year program that will result in a Bachelor Degree in Computer Science and Masters in Curriculum and Instruction. Because the fifth year studies will match the MCERT (having much overlap of the educational components of the CS Education Track), these students will be expected to pursue the Traditional Track in CS as undergraduates. However, by making application to the joint program mid-way through their undergraduate degree, they will be allowed to take (and count to both degrees) up to 12 credits of their MCERT work as part of their undergraduate study. (Students who apply to the MCERT later will not be exempt from these credits; they will need to satisfy all credit obligations typical of any other teacher who is returning for MCERT credentials.)

Requirements for Computer Science Bachelor Degree – Education Track
The course of study for this track must include all of the following requirements with a grade of C or higher in each of the following courses:

1. Computer Science Courses
    a. CMSC 131 or a score of 5 on A version of the JAVA Advanced Placement exam or a score of 4 or 5 on the AB version of the JAVA Advanced Placement exam or an acceptable score on the appropriate Department exemption examination, which is to be taken at the time of entry into the program. [Drafting note: There is no AB exam anymore, so even the original CS track requirements will need updating. We leave the language here the same as for the original track in order to remain consistent, with the expectation that all will be updated together.]
    b. CMSC 132 or acceptable score on the appropriate Department exemption examination, which is to be taken at the time of entry into the program.
    c. CMSC 250 or acceptable score on the appropriate Department exemption examination, which is to be taken at the time of entry into the program.
    d. CMSC 216 or acceptable score on the appropriate Department exemption examination, which is to be taken at the time of entry into the program.
    e. CMSC 351, CMSC 330, CMSC 420 and CMSC 411
    f. At least two of CMSC 424, CMSC 433 and CMSC 434
    g. Either one additional CMSC 400-level elective (at 3 credits or more) or three of the 1-credit CMSC elective offerings at 3/400-level.
    h. CMSC 4XX-Diversity in Computer Education
    i. CMSC 4YY-Computer Education Project Development

2. Supporting Education Courses
    a. EDHD 413 (Adolescent Development)
    b. EDHD 426 (Cognition and Motivation in Reading)
    c. EDPS 210 or EDPS 301: Issues in Education
    d. EDCI 463: Reading in the Content Areas
    e. EDCI 4XX-Methods I
    f. EDCI 4YY-Methods II
    g. EDCI 3xx Field Experience
    h. EDCI 474

3. Supporting Math Courses:
    a. MATH 140 and 141.
    b. MATH 240.
    c. A STAT course which has MATH 141 (or a more advanced mathematics course) as a prerequisite. Most commonly this would be STAT400.

Requirements for the Double Major in Secondary Education

The CS Education Track requirements listed above will also encompass all that is necessary for certification except for student teaching and its related seminar, which are typically obligations that a major in any of the science-education tracks would satisfy in his or her final semester (as all other classroom experiences must happen first.) Operationally, the field experience is a time when only those classes are taken, because of the student's near-full time obligation out on site.

In order to obtain the Double Major in Secondary Education, the CS Education Track major must complete EDCI 4TT (Student Teaching) and EDCI 4SS (Student Teaching Seminar), and satisfy the additional GPA requirements (as below.) The full expectation is that students entering the CS education track are intent on the double major; nevertheless, the student teaching (segregated to the end of the undergraduate experience) is not considered part of the stand-alone CS degree, as this will recognize that by the time majors reach this point, it may already be apparent that their prospects for the double major may be bleak. This proposed arrangement thus provides a student with a reasonable way to complete bachelor requirements (for the stand-alone degree, even of one that is un-credentialed) without having to backtrack through a host of Traditional Track CS requirements (which, by adding a year or more, could make a degree unreachable for many.)

Additional GPA Requirement

Students choosing to complete the Computer Science Secondary Education Track with Double Major in Secondary Education must maintain a GPA of 2.75 overall. Since the College of Education (a partner in this program) has Professional Education Admission Program (PEAP) status, it is already appropriate that no student with a GPA lower than 2.5 be able to remain in the Computer Science Education Track. The higher standard reflects our interest in excellence and is intended to ensure that students will be able to enter a master's certification program in Education at some later point. For the five-year program the GPA requirement is 3.0 for other programs in the College of Education, and those students will need to conform to those standards also.

Learning Outcomes: Graduates of the Computer Science Education Track will

1. understand data representation, abstraction and application to problem solving in multiple programming paradigms. These skills will be developed through a sequence of programming courses in multiple languages,
2. be able to create, augment, debug and test computer software in a variety of development environments. These skills will be built progressively through the courses in the introductory sequence of courses,
3. develop mathematical and analytical reasoning skills,
4. experience design and implementation of programming projects that are more similar to those that would be seen in a real world environment using a wide variety of contexts and environments,
5. manifest collaborative human interaction skills in several ways: design of software/hardware based on user input and feedback, working as a member of a programming team, and making presentations to groups about what has been designed and/or implemented,

6. obtain a high level of skill in mathematical reasoning about algorithms and data structures and other objects in computer science,
7. demonstrate skills and understanding relative to social aspects of computing that are appropriate for specialists and non-specialists including professional organizations, educational technology and emerging technologies,

In addition, graduates with a double-major in secondary education will

8. demonstrate knowledge of human development, psychology, diversity and pedagogy to be effective teachers of computer science at the middle and high school levels.

New Courses

Implementing all of these programs will require four new courses, as attached. (Two are CMSC courses, two are EDCI methods courses.)

Computer Science Secondary Education Sample Four-year Plan

NOTE: This is a proposed plan and the College of Education does not guarantee that these courses will be offered in the designated semester. Consult the Schedule of Classes for class availability and meeting times.

FRESHMAN YEAR:

Fall Semester:	Cr.	Spring Semester:	Cr.
CMSC 131	4	CMSC 132	4
MATH 140 (MS)	4	MATH 141	4
ENGL 101 (FE)	3	CORE Literature (HL)	3
CORE SH	3	CORE SB	3
Total Credits	14	Total Credits	14

SOPHOMORE YEAR: Apply to the College of Education between 45 and 60 credits

Fall Semester:	Cr.	Spring Semester:	Cr.
CMSC 250	4	CMSC 330	3
CMSC 216	4	CORE SB	3
EDPS 210 (satisfies CORE HO)	3	CORE Arts (HA)/Diversity (D)	3
CORE PS/LS	3	CORE-PL/LL	3
CMSC Elective (language course)	1	CMSC 351	3
Total Credits	15	Total Credits	15

JUNIOR YEAR:

Fall Semester:	Cr.	Spring Semester:	Cr.
CMSC 420	3	Reading I - EDHD 426	3
Adolescent Development - EDHD 413	3	ENGL 391/393	3
CS Methods I - EDCI 4XX	3	CMSC 4XX (Diversity in Computer Education)	3
CMSC 411	3	EDCI 4YY (Methods II)	3
MATH 240	4	CMSC 424	3
		CMSC Elective (language course)	1
Total Credits	16	Total Credits	16

SENIOR YEAR:

Fall Semester:	Cr.	Spring Semester:	Cr.
CMSC Elective (language course)	1	EDCI 4TT Student Teaching	12
STAT 400	3	EDCI 474	2
EDCI 463 Reading II	3	EDCI 450 Student Teaching Seminar	1
CMSC 4YY (CMSC project development)	3		
CMSC 433	3		
EDCI 3xx Field Experience	1		
Elective	1		
Total Credits	15	Total Credits	15

NOTES about courses to be developed are on the reverse side.

Total Credits: 120

Notes:

1. EDCI 4XX will be a course about teaching but the content will be based on the computer science courses already taken. 3 credits. Prereq: CMSC 330 (so there has been exposure to several languages and areas of computer science). The exercises will center on developing lesson plans, managing groups and writing student exercises and assessments.
2. EDCI 4YY will be a course applying the skills discussed in EDCI 4XX with a further level of abstraction to whole course and program design. 3 credits. Prereq EDCI 4XX. The whole process of program and course design as well as expectations and accreditation will be discussed. This course will be more project based than EDCI 4XX, and the project descriptions and implementations will be critiqued by the group.
3. CMSC 4XX will be a course on diversity in learning, society, ethics and Emerging Issues in Computer Science. 3 credits. Written assignments on these issues and use of electronic teaching tools such as presentation software, wiki and on-line quiz development tools. The emphasis will be on the variety of learning outcomes in computer science course and the variety of resources available to reach students of different backgrounds and interests. This course will be limited to those in the computer science education track.
4. CMSC 4YY will be a new projects based course intended to give a tour of emerging technology areas and the tools needed to build accessible applications in them. The focus will be on organizing small teams on modest sized projects across a whirlwind sequence of areas on robotics, graphics, forensics, communication networks, remote sensing and multi-media systems, chiefly driven by scripting languages. Covers proposal development through presentations of prototype developed with analysis of initial description, to product implementation. This course will be limited to those in the computer science education track. Prereq CMSC420. 3 credits.

MCERT for Computer Science

Summer (June - August):

CMSC 4XX (Diversity in Computer Science)	3
Adolescent Learning - EDHD 413	3
Reading Course 1 - EDHD 426	3
Embracing Diversity in the Classroom - EDCI 697	3
Total Credits	**12**

Fall (September-December)

Methods I - EDCI 4XX	3
CMSC 4YY (CMSC Project Development)	3
Conducting Research on Teaching EDCI 698	1
Teaching Internship	5
Total Credits	**12**

Spring (January - May)

Methods II - EDCI 4YY	3
Reading Course 2 - EDCI 763	3
Conducting Research on Teaching EDCI 698	2
Teaching Internship	4
Total Credits	**12**

Summer (May - June)

Teaching as a profession EDCI 690	3
Teaching Internship	3
Total Credits	**6**

Total Credits: 42

MCERT Admissions Guide for computer science content material requirements

Course(s) where this experience was gained	Description of minimum experience in specific areas of computer science.
	1. Significant programming experience a. at least a two semester sequence in one language b. programs written in increasing difficulty through the sequence
	2. Comparative Programming Languages or overview a. at least one course on the variety of families of languages b. including the overall concept of programming language structure
	3. Proofs and Analysis a. at least a two semester sequence in proof methods and algorithm analysis b. proofs of runtime efficiency
	4. Data Structures and Control a. at least one course that emphasizes larger dynamic data structures b. linked lists, binary trees, hashing structures, graphs
	5. Computer Architecture a. at least one course that emphasizes architecture and/or assembly programming
	6. Exposure to a variety of areas in computing a. at least three additional courses at the 300/400 level on special topics b. Networking, Compilers, Software Engineering, Databases, Artificial Intelligence

Computer Science Secondary Five-year Plan

NOTE: This is a proposed plan and the College of Education does not guarantee that these courses will be offered in the designated semester. Consult the Schedule of Classes for class availability and meeting times.

FRESHMAN YEAR:

Fall Semester:	Cr.	Spring Semester:	Cr.
CMSC 131	4	CMSC 132	4
MATH 140 (MS)	4	MATH 141	4
ENGL 101 (FE)	3	CORE Literature (HL)	3
CORE SH	3	CORE SB	3
Total Credits	**14**	**Total Credits**	**14**

SOPHOMORE YEAR: Apply to the College of Education between 45 and 60 credits

Fall Semester:	Cr.	Spring Semester:	Cr.
CMSC 250	4	CMSC 330	3
CMSC 216	4	CORE SB	3
EDPS 210 (satisfies CORE HO)	3	CORE Arts (HA)/Diversity (D)	3
CORE PS/LS	4	CORE-PL/LL	3
		CMSC 351	3
Total Credits	**15**	**Total Credits**	**15**

JUNIOR YEAR:

Fall Semester:	Cr.	Spring Semester:	Cr.
Elective	3	Reading I - EDHD 426	3
Adolescent Development - EDHD 413	3	ENGL 391/393	3
Elective	3	CMSC 4XX (Diversity in Computer Education)	3
CMSC 411	3	CMSC 420	3
MATH 240	4	Elective	3
		CMSC Elective (language course)	1
Total Credits	**16**	**Total Credits**	**16**

SENIOR YEAR:

Fall Semester:	Cr.	Spring Semester:	Cr.
EDCI 463 Reading II	3	STAT 400	3
Embracing Diversity in the Classroom - EDCI 697	3	CMSC Elective (language course)	1
EDCI 3xx Field Experience	1	Elective	3
Elective	3	CMSC 433	3
CMSC 424	3	EDCI 474	2
CMSC Elective (language course)	1	Elective	3
Elective	1		
Total Credits	**15**	**Total Credits**	**15**

Graduate YEAR:

Fall Semester:	Cr.	Spring Semester:	Cr.
CS Methods I - EDCI 4XX	3	Reading Course 2 - EDCI 763	3
Conducting Research on Teaching - EDCI 698	1	Conducting Research on Teaching EDCI 698	2
Teaching Internship	5	Teaching Internship	4
CMSC 4YY (CMSC project development)	3	CS Methods II - EDCI 4YY	3
Total Credits	**15**	**Total Credits**	**15**

Summer:

Summer (May -June)	Cr.
Teaching as a Profession	3
Teaching Internship	3
Total Credits	**15**

Total Credits: 150

NOTES about courses to be developed are on the reverse side.

CMSC 4XX - Diversity in Computer Education

Note: This is an example content-centric syllabus. Current University policies on excused absences, academic honesty, and course-specific policies would be added for an actual syllabus distributed to the students.

Course Description

This course will cover the history and emerging issues in computer science to make students aware of the societal and ethical issues both in the computer science content area and in the teaching of computer science. It demonstrates how computing tools can be used to address the diversity of learners in the classroom and discusses the variety of pedagogical approaches available to ensure that all students are learning the material.

Learning Outcomes

- Understand history behind the technology in use today including the diversity of development paths and diversity of people
- Understand how computer applications can be used to assist students - both computer adaptive technologies for physical limitations and computer assisted instruction software for variety in the learning environment
- Understand how societal events influenced the development of computers both positive and negative, how professional organizations and standards have affected that development
- Have a basic understanding of international effects of technology development and proliferation
- Understand the current state of computing and the areas of society influenced by its existence
- Understand the ethical issues around computers and their applications and considerations concerning computer use in education

Assessments

Semester Exam 1 (15%)
Semester Exam 2 (15%)
Final Exam (28%)
Projects – (14%)
Written Homework (28%)

Content outline

- Understanding the students and their individual needs especially how technology can be used to address special needs
- How computers have changed society and how the needs of society have influenced computer development
- Current state of the application of computers in education
- Current state of adaptive technologies and educational technology
- Arguments around content of computer science courses at the secondary school level
- Computer Ethics and student safety standards - for suggested content see: http://bibliotecavirtual.clacso.org.ar/ar/libros/raec/ethicomp5/docs/pdf_papers/43Martens,%20Bern.pdf and http://www.springerlink.com/content/57n67v8161270665/
- Different styles of computer assisted instruction including the benefits and drawbacks of those application
- Underrepresented populations in STEM (women and minorities) along with discussions of learning styles, individual goals and motivations
- Information about professional organizations and teaching standards
- Web and Internet site evaluation and safety

Suggested Projects and written assignments

- Development of on-line quizzes using a quiz creation tool including making it available on the web
- Use of tools such as a wiki for class communication with discussion of the benefits and drawbacks of using that tool
- Individual/group presentations of computer ethics case studies - students use power point and/or movie maker to share the individual case studies for discussion with the group
- Individual research papers on specific areas of history, student populations and/or internet safety

CMSC 4YY - Computer Education Project Development

Note: This is an example content-centric syllabus. Current University policies on excused absences, academic honesty, and course-specific policies would be added for an actual syllabus distributed to the students.

Course Description

A project-based course intended to give a tour of emerging technology areas and the tools needed to build accessible applications in them. The focus will be on organizing small teams on modest sized projects across a broad sequence of areas on robotics, graphics, forensics, communication networks, remote sensing and multi-media systems, some of which can be chiefly driven by scripting languages.

Learning Outcomes

- Gain exposure to multiple emerging areas and gain experience implementing projects in these different areas
- Understand the proposal and design phase of project development for computer science projects
- Gain understanding of group dynamics by developing these projects as a team
- Gain presentation experience by sharing what you have learned about your specific area of computer science with the other members of the class
- Learn feedback techniques by giving other groups feedback on their presentations

Assessments

Semester Exam 1 (15%)
Semester Exam 2 (15%)
Final Exam (28%)
Projects – (28%)
Written Homework (14%)

Content outline

- Understand presentation style options
- Understand group work management - its benefits and its drawbacks
- Brief exposure to many of the following topics and, in addition, to any new developments in computer applications: robotics, graphics, forensics, communication networks, remote sensing and multi-media systems, chiefly driven by scripting languages
- Practice implementations and presentations on multiple topics areas of emerging computing applications

Suggested projects and written assignments

- Project proposal/design presentations
- Analysis/critique of proposal/project description by the other groups
- Project development
- Project prototype presentations

EDCI 4XX - CS Methods I

Note: This is an example content-centric syllabus. Current University policies on excused absences, academic honesty, and course-specific policies would be added for an actual syllabus distributed to the students.

Course Description

This course introduces methods for teaching computing at the secondary level including the teaching of programming, computer literacy and computational thinking. It covers project development, written assessment development, grading rubric development and presentation these teaching materials.

Learning Outcomes

- Understand current argument concerning the content for secondary school computing classes.
- Understand presentation and classroom exercise options used to increase student understanding of that material.
- Understand the methods of project creation and grading.
- Understand the methods of written assessment creation and grading.

Assessments

Semester Exam 1 (15%)
Semester Exam 2 (15%)
Final Exam (28%)
Projects – (14%)
Written Homework (28%)

Content outline

- Practice computer science content presentation skills.
- Design group exercises and understand how to teach management skills.
- Practice writing descriptions for student project assignments - the special challenges of projects for individuals and for groups.
- Understand and practice test and quiz development.
- Understand and practice grading rubric development and application.

Suggested class exercises, projects and written assignments

- Class exercises: example - topic of abstraction (expanded version available in Inroads - volume 25:4, p. 32)

o Discussion about the definition of abstraction as understood from CS classes
o Students design activities on how to teach the concept of abstraction
o Students analyze to determine what features of abstraction are most difficult to teach
o Discussion about what computer science topics are similar or can be taught in similar ways
o Design of projects and exam questions that would reinforce the concept of abstraction for students

- Writing and critiquing project descriptions - the students implement small projects as described by a written description to see the level of detail that must be present and/or issues raised from lack of or too much detail
- Written exercise about methods of assigning/managing group work in the high school classroom. Presentations and discussions about the different methods addressed in the written exercise.

EDCI 4YY - CS Methods II

Note: This is an example content-centric syllabus. Current University policies on excused absences, academic honesty, and course-specific policies would be added for an actual syllabus distributed to the students.

Course Description

This course will further prepare students for teaching the topic of computer science primarily through the presentation of more options and the practice of preparing complete lesson plans and the philosophical basis of what should be included in those plans and evaluation of student work on assessments in those plans. It includes laboratory sections to allow for programming exercises and evaluation of them as teaching tools.

Learning Outcomes

- Understand the philosophical bases for lesson planning.
- Understand how to organized topics for maximum retention of material.
- Practice creating plans, assessments, and rubrics.
- Practice grading and giving constructive feedback.

Assessments

Semester Exam 1 (15%)
Semester Exam 2 (15%)
Final Exam (28%)
Projects – (28%)
Written Homework (14%)

Content outline

- Review literature on types of lesson plans and create plans based on those different sets of criteria.
- Practice writing assessments and their corresponding grading rubrics as they would fit into a lesson plan.
- Discuss lesson planning at many different levels: the curriculum, the course, the individual lesson and the individual activity.
- Grading rubric development including application of the rubric to individuals and groups in different types of assessments.

Suggested class exercises, projects and written assignments

- Class exercises: example - exam writing (expanded version available in Inroads - volume 25:4, p. 31)
- Discuss test design: the target and structure of the test, different ability levels of students, types of questions, different complexity levels of questions, matching questions to the class level, organization of the questions in the test, and grading policies.
- Students work in teams to construct questions of different levels and types for a specific topic.
- Groups exchange to take the test (not always writing the correct answer but making sure the questions are understandable)
- The papers are exchanged back so the grading rubric can be applied.
- Class exercises: example - programming projects - take the same approach as above described for an exam and apply it to a short programming project.
- Discussion of curriculum/program assessments and accreditation and the corresponding curriculum provided by accrediting organizations - comparisons.

Mapping the Computer Science Educator Standards to the UMCP Computer Science Education Track

There are several different organizations publishing information about computer science education. One of these is of these is the Association of Computing Machinery (ACM), specifically the Special Interest Group on Computer Science Education (ACM-SIGCSE).[1] Another is the Computer Science Teacher's Association (CSTA) which has several reports on the current state of accreditation and of qualifications of computer science teachers.[2] A third is the International Society for Technology in Education (ISTE) and their National Education Technology Standards (ISTE-NETS).[3] And the fourth organization contributing to this area is the National Council for Accreditation of Teacher Education (NCATE).[4]

The following quote appears on the NCATE web page referring programs for computer science teacher education to the ISTE standards.

> The 2002 edition of ISTE Standards for Computer Science Education are applicable to program reports submitted through 2011. ISTE is submitting a new set of standards to NCATE in Fall 2011. If approved, programs can use either set of standards through Spring 2013. Beginning in Fall 2013, programs submitting reports must use the new set of standards.
> [http://www.ncate.org/Standards/ProgramStandardsandReportForms/tabid/676/Default.aspx - retrieved August 1, 2011]

The standards on the ISTE web site were originally only about the application of technology to education, but they have expanded those standards to include requirements for different areas of teaching technology itself.

On the ISTE they present these updated for Computer Science Educators. On this web page are descriptions of standards for several areas that include computing resources applied to education. Each of these are called NETS and the web page include variations - NETS for teachers, for students, etc.

> ISTE has refreshed the original NCATE standards--formerly known as the Technology Leader,Technology Facilitator, and Secondary Computer Science Educator standards. These new standards are meant to be used with the other NETS to guide transformation of our schools. They can be used for professional development, evaluation, and hiring. ISTE will provide resources to support implementation of these new standards soon!
> [http://www.iste.org/standards.aspx - retrieved August 1, 2011]

[1] ACM-SIGCSE: http://www.sigsce.org

[2] CSTA: http://csta.acm.org/Communications/sub/DocsPresentationFiles/CertificationFinal.pdf
http://csta.acm.org/ComputerScienceTeacherCertification/sub/TeachCertRept07New.pdf
http://csta.acm.org/Research/sub/Projects/ResearchFiles/TeacherSciencePrep.pdf
http://csta.acm.org/Research/sub/Projects/ResearchFiles/StateofCSEDHighSchool.pdf
http://csta.acm.org/ComputerScienceTeacherCertification/sub/TeachCertRept07New.pdf
http://csta.acm.org/includes/Other/CSTAStandardsReview2011.pdf

[3] ISTE-NETS: http://www.iste.org/standards.aspx
http://www.iste.org/Libraries/NETS_Refresh_Toolkit/NETS_for_Computer_Science_Educators.sflb.ashx

[4] NCATE: http://www.ncate.org/Standards/ProgramStandardsandReportForms/tabid/676/Default.aspx

This document will present the correlation that exists between the the ISTE NETS for Computer Science Educators (NETS•CSE) to the courses required in our proposed computer science education program. The list of content areas is available in PDF format on the ITSE website.[5] These standards are divided into areas of Knowledge of Content, Effective Teaching and Learning Strategies, Effective Learning Environments and Effective Professional Knowledge and Skills.

1. **Knowledge of Content.** Computer Science Educators demonstrate knowledge of Computer Science content and model important principles and concepts.

The first subcategory in this area states that teachers need to be able to demonstrate knowledge of and proficiency in data representation and abstraction. In the Object Oriented Programming I course (CMSC 131), the students learn to use the primitive data types of the Java Programming Language. The course content included static and non-static class data members as well as variables that are local to methods. All of the primitive data types available in Java are presented in this class and projects are developed by the students to use all variations of the primitive data types. This understanding and application of different primitive data types is extended into several other languages in the Organization of Programming Languages course (CMSC 330). In that course, several different families of languages are compared in such a way that students can see how each family of languages manages its data. The students have the opportunity to apply the primitive data of several different languages through the projects in that course. The understanding of data is approached from a slightly different angle in the Introduction to Computer Systems course (CMSC 216). In that course, the students learn to use the primitive data types of C, but they also see how the data is actually stored on the computer system by programming in assembly and doing bit level manipulation of that data. This approach allows a deeper understanding of primitive data types because the abstraction is removed and students see and manipulate the bits themselves.

The second subcategory states that students should demonstrate an understanding of static and dynamic data structures. This topic is also presented in several courses using several programming languages. In the Object Oriented Programming I and II courses (CMSC 131 & 132), students use both static and dynamic data structures in Java. These include arrays, linked lists, stacks, queues, trees, graphs, hash tables and maps. The correct use of each of these structures is presented in class and students have the opportunity to practice with a selection of these data structures through the projects they implement. In the Introduction to Computer Systems (CMSC 216) course, the students continue the use of both static and dynamic data structures through application in the C programming language. The students implement projects that include a selection of the following structures: both static and dynamically allocated arrays, linked lists, hash tables and trees. Many of the languages presented in the Programming Languages course (CMSC 330) also introduce data structures implemented in each of those specific languages. The most comprehensive presentation of the more advanced dynamic data structures occurs in the Data Structures course (CMSC 420), which is also required in this track. In that course more advanced forms of trees as well as several other structures such as skip lists are introduced and implemented in projects. The other side of data structures - the analysis of

[5] http://www.google.com/url?sa=t&source=web&cd=1&ved=0CCQQFjAA&url=http%3A%2F%2Fwww.iste.org%2FLibraries%2FNETS_Refresh_Toolkit%2FNETS_for_Computer_Science_Educators.sflb.ashx&rct=j&q=NETS_for%20computer_science%20education&ei=CP9hTonuDZLegQexr-STCg&usg=AFQjCNEPsQY_6WQpHMPcF-TgKH7-QXwLuQ&sig2=9Jxfjz9cwo7zOADfSnaAT- retrieved August 1, 2011.

the algorithms for processing data stored in them - is presented and practiced in the Algorithms course (CMSC 351). In this course they look at several of the same data structure processing algorithms that were introduced in the other courses, but in this course they learn the methods needed to analyze those algorithms in order to make intelligent decisions both about what data structure to apply and what algorithm is best for the given problem.

The third subdivision states that students should effectively use, manipulate, and explain various external data stores - various types (text, images, sound, etc.), various locations (local, server, cloud), etc. The projects required in the required programming sequence of courses all use various types of data: there are projects about manipulation of photographs and others about manipulation text; for example, there are projects that require input from graphical interfaces and projects that require text input; and there are projects produce textual reports and others that produce pictures. Because of the variety of language used, the students also get to see and apply their programming to a variety of data locations; for example: inter-process communication projects are done in both Java and C. All of the material needed for these projects are covered in lecture, presented in the textbook and electronic sources, practiced in the lab/recitation and then applied by the student in the longer term programming projects. This leads to a deep understanding of a wide variety of data storage.

The last subdivision in this section states that computer science educators must be able to effectively use modeling and simulation to solve real-world problems. In all of the required programming classes, most of the projects are presented as problems to be solved. In the beginning courses these problems are simplified beyond what would occur in the real-world, but as the courses advance, the projects get closer to solving real-world problems. From the beginning they are encouraged to use modeling and simulation techniques to develop and test their solutions to these problems.

The students in this program are exposed to a wide variety of languages, data and problem solving strategies all designed to make them able to more efficiently and effectively apply computers to solve problems. The wide variety of languages and data not only gives them the ability to apply those, but also allows them to adjust to new languages and new ways of storing data are developed in the future.

2. **Effective Teaching and Learning Strategies.** Computer Science Educators demonstrate effective content pedagogical strategies that make the discipline comprehensible to students.

Through the lower level courses, students will learn the content of computer science as described above, but in this track their will need to be courses that specifically assist the students in developing teaching skills and help them to develop an understanding of the students they will be teaching. In order to accomplish this, the computer science department will have two courses that are specifically geared toward education. In the proposal, these are numbered as CMSC 4XX and CMSC 4YY. There will also be two courses in the College of Education to support the students' understanding of pedagogy and to allow them to practice the related skill set. These education courses are numbered on the plan as EDCI 4XX and EDCI 4YY.

The first of these classes, EDCI 4XX is titled CS Methods I and its main purpose is to introduce the student to methods for teaching computing at the secondary level including the teaching of programming, computer literacy and computational thinking. It will help the students develop the skills needed to create good assessments - both projects and written assessments, the skills needed to create good grading rubrics and the skills needed to present material. Topics will include those that help the future teachers to identify areas that may be problematic for their students and how to develop projects that allow students to practice and therefore overcome those difficulties. The projects need to also be designed so that they are developmentally appropriate and span interest areas in order to appeal to a larger portion of the population. The benefits of collaborative group work on some of the projects will also be discussed with readings about how groups can best be constructed and utilized. Written assessment development and grading rubric development will be also be discussed in such a way as to allow the future teachers to practice different methods of presentation by presenting their ideas to their peers in such a way as to gain feedback from the other members of the class. One example of a class exercise would be to look at a single computer science topic, have students develop a lesson plan on how to present that concept to students, develop exercises and exam questions that would reinforce the lesson, present their ideas on that concept to the class and receive feedback from their peers on how that one topic. Since this class is taken after the students have completed several computer science courses, there will be a wide range of possibilities for the topics to be taught.

The first education class in the computer science department is on the proposed schedule as CMSC 4XX and is titled Diversity in Computer Education. This class would cover the history and emerging issues in computer science and make the students aware of the societal and ethical issues inherent in computer use. The course would look at history both through the lens of how computers developed but also how the use of computers affected societal development over time. Most specifically, the course would look at how education has been changed by the inclusion of technology and how this can allow more variety of learning styles that engage and empower learners from diverse cultural and linguistic backgrounds. One project could be to assess availability of tools and practice using an on-line quiz creation tool to develop additional practice for their students. The application of computer-assisted instruction can help capture the interest of some students and provide remediation and extra practice for others. Understanding the variety of student motivations and backgrounds will help these teachers know how to reach the underrepresented populations and how to direct lessons to reach those goals and motivations that are individual to each student.

The second computer science education course from the College of Education is CS Methods II and is shown as EDCI 4YY on the proposed plan. In this course, the student will gain more experience in presentation and in preparing of lesson plans, assessments and rubrics. This course will include comparisons of different styles of lesson planning and assessment writing. One activity would be to develop a short exam on some small topic in computer science and its grading rubric. The exam would then be taken by the other students in the class and graded according to the rubric. The results and the reflection on this assessment would then be presented to the class in order to gain feedback about how well the assessment would fulfill the needs of a certain audience of students. Since this course is taken after several computing courses have been completed, there are ample topics to choose from for the exam writing and grading exercise. Seeing the different types of assessments created by classmates and discussing the benefits and drawbacks of each will allow the future teach to learn how to create multiple forms of assessment and to use the resulting data to capture student learning and shape classroom instruction.

The second course for computer science education from the Department of Computer Science is on the schedule as CMSC 4YY and is titled Computer Education Project Development. Since the application of the computer to problem solving is the focus of what must be taught in the high schools, a course where the project development aspects are the focus is essential. The course will be a project based course intended to give a tour of emerging technologies and the tools needed to build accessible applications in them. The focus will be on organizing small teams on modest sized projects across a whirlwind sequence of areas on robotics, graphics, forensics, communication networks, remote sensing and multi-media systems. The students will learn about group dynamics and how to create exercises that will engage the students because of their applicability to real-world problems. They will learn how to encourage innovative thinking and true problem solving by reflecting on their own reasoning process as they develop projects. They will improve their presentation skills as they practice giving presentations both within their group and as a group to the remainder of the class. Through critical analysis of the projects of the other groups, they will learn to identify problematic constructs in computer science and through meta-cognition on the learning needed for their own project, they will develop appropriate strategies for helping others overcome difficulties. In addition to the project development itself and the presentations of that project, a major focus of this course will be the reporting on what was learned and how it was learned through the project development process.

3. Effective Learning Environments. Computer Science Educators apply their knowledge of learning environments by creating and maintaining safe, ethical, supportive, fair, and effective learning environments for students.

In all of the computer science classes, the safe and effective use of computer hardware, software, peripherals and networks is demonstrated and the dangers of improper use are discussed. In the new courses designed specifically for Computer Science Education (CMSC 4XX and CMSC 4YY), the proper use will be discussed along with the computer safety and ethics issues. For these classes, the discussion has to go beyond just demonstrating to how these skills can be demonstrated to middle school and high school students. This would need to include what types of rules should be in place to ensure engaging learning in an equitable and accessible classroom.

4. Effective Professional Knowledge and Skills. Computer Science Educators demonstrate professional knowledge and skills in their field and readiness to apply them.

In the Diversity in Computer Science Education course (CMSC 4XX), the wide range of course content taught as computing in the middle schools and high schools will be discussed. This will also include information about the professional organizations and the teaching standards currently in place on the local, state and national level. These current trends in computer science as it affects both education and society in general will be discussed in different ways in the computer science education courses. The discussion in the Diversity class about the history of computing and its affect on society, will naturally flow into the current interactions between computers and society. In the Computer Education Project Development course (CMSC 4YY), the projects will be based on the new trends in computer science and therefore how these new developments would affect education will be a primary focus.