
Subject: FW: FW: Proposal for a Minor in Computational Science
Attachments: SampleCoursePlans.pdf

From: P Sadayappan [mailto:psaday@gmail.com]
Sent: Tuesday, February 02, 2010 5:47 PM
To: Collier, Alexis
Cc: sadayappan.1@osu.edu; Soave, Melissa
Subject: Re: FW: Proposal for a Minor in Computational Science

Dear Professor Collier:

- 1) We had not earlier thought about the issue of minimum number of courses/hours that a student must be required to complete at Ohio State. A restriction can be imposed that a student may take no more than two (or perhaps three) courses from other institutions.
- 2) The attached sheet shows three sample course plans, for students in Computer Science, Materials Science, and Mechanical Engineering. As can be seen, only two to four total courses will really be needed to complete the requirements of the computational science minor (since course required for the major can be double-counted for a minor in the College of Engineering).

I would be happy to come over to Bricker to answer any further questions that the committee may have.

Regards,
-- Saday

On Mon, Jan 25, 2010 at 5:07 PM, Collier, Alexis <collier.1@osu.edu> wrote:

Dear Professor Sadayappan:

Thank you for talking with me today about the Computational Science Minor proposal that was recently reviewed by the Council on Academic Affairs. I apologize for the delay in providing you feedback about the Council's remaining questions. In brief, the Council has asked for the following additional information:

1. What are the requirements for the minimum number of hours that a student must complete from the OSU course list in order to receive the minor from OSU (versus courses which can be completed online from other institutions)? Could, for example, a student receive a minor from OSU without having taken any OSU courses?
2. Provide information / examples of typical paths students might take in order to complete the minor.

The issue regarding the second question is that courses in the minor have a large number of pre-requisites, and the Council wanted information as to how students would actually complete the minor. Would, for example, the requirements + pre-requisites make it likely that only students from certain majors in Engineering take it? This may be acceptable, but they wanted to be clear about any such functional restrictions. A table showing sample minors and courses/hours taken, including pre-requisites, would be helpful.

Additional detail about the discussion may be found in the December 2, 2009 minutes posted at: http://oaa.osu.edu/caa/documents/CAAM091202_001.pdf.

If you are able to prepare the responses and get them to me by early next week, I can share them with the Council at the February 3, 2010 meeting (3:00-5:00 p.m., 200 Bricker). Depending on the response, I would appreciate it if you could hold the next two meeting dates, February 3 and 17, in case the Council would like to talk with you again.

Thank you very much. Please feel free to contact me if I may answer any of your questions.

Alexis

Alexis C. Collier, Ph.D.
Associate Provost
The Ohio State University
203 Bricker Hall
190 North Oval Mall
Columbus, OH 43210-1321
Voice: 614-292-5881
FAX: 614-292-3658

From: mendelsohn.1@mecheng.ohio-state.edu [mailto:mendelsohn.1@mecheng.ohio-state.edu] **On Behalf Of** mendelsohn.1@osu.edu
Sent: Wednesday, July 15, 2009 9:07 AM
To: sadayappan.1@osu.edu
Cc: Smith, Randy; Collier, Alexis; Soave, Melissa
Subject: Proposal for a Minor in Computational Science

Hi Saday,

I have read again your proposal and, while the rationale, goals/outcomes, administration/advising, and the course choices of the minor are extremely well laid out, I still have a few questions/requests now that it is before CAA.

1. In the concurrence letter from Mechanical Engineering (that I wrote wearing another hat) I mentioned that having ME 785 alone might be a problem because ME 784 is a true pre-requisite. You should either include them both or be up front that ME 784 is a pre-req for ME 785. I think that ME 785 is quite appropriate for the minor, but they really need ME 784 first, for which we will admit students by permission of instructor.

2. It would be very nice to have a one-page information sheet from a student's perspective that summarizes the minor and includes information about (a) number of hours required, (b) the rules about double-counting of hours with the GEC or the major requirements. In the College of Engineering that rule is that double counting with the GEC is allowed and that double counting with the major is up to the major department. You mention that there is a 20 hr minimum, but the maximum of 30 hrs for the College of Engineering is not mentioned.

3. To go along with the information page mentioned above, it would also be nice to have several sample minor programs laid out including an hour count. With the 7 course requirement (six areas and an elective) I am wondering how easy/probable it is that students can complete the minor in less than the maximum hrs of 30 allowed. This is about truth in advertising too. You should be up front about the number of hours required for the "typical" path through the minor. The 20 hr minimum alone is a bit misleading I think. Besides the hours issue a few sample paths through the minor would help prospective students see what a coherent program looks like.

Please feel free to contact me if you want to discuss these issues.

Thanks,

Dan Mendelsohn, Chair
Council on Academic Affairs

Sample course plans for a minor in Computational Science

Sample program for a student from Computer Science & Engineering

Requirement	Course	Required for Major?	Prerequisites
Sim&Modeling	CSAC 245 (Capital)	No	None
Prog&Algorithms	CSE 221	Yes	
Num Methods	CSE 541	Yes	
Optimization	COMP 345 (Wittenberg) CSE 778	No for HW/SW Option	None
Capstone	CSE 699	No	CSE 621
Domain-Course	CSE 680	Yes	
Elective	Math 255/415/568/571 CSE 621	No No (Tech. Elective)	Math 254 (Rqd by Major)

Total number of courses required for minor, beyond inherent requirements for CSE degree: 4

Sample program for a student from Materials Science & Engineering

Requirement	Course	Required for Major?	Prerequisites
Sim&Modeling	MSE 533	Yes	
Prog&Algorithms	EG 167	Yes	
Num Methods	ME 250	No	Math 415 (Rqd by Major)
Optimization	MSE 600	Yes	
Capstone	MSE 695	Yes	
Domain-Course	MSE 756	No	Perm. Instr.
Elective	Math 415	Yes	

Total number of courses required for minor, beyond inherent requirements for MSE degree: 2

Sample program for a student from Mechanical Engineering

Requirement	Course	Required for Major?	Prerequisites
Sim&Modeling	CSAC 245 (Capital)	No	None
Prog&Algorithms	CSE 294P or EG 167	No	None
Num Methods	ME 250	Yes	
Optimization	COMP 345 (Wittenberg) (alternate) ME 761	No No	None ME 562 (Rqd by Major)
Capstone	ME 564	Yes	
Domain-Course	ME 639	No	ME 420 (Rqd by Major)
Elective	Math 415	Yes	

Total number of courses required for minor, beyond inherent requirements for ME degree: 4



College of Engineering

122 Hitchcock Hall
2070 Neil Avenue
Columbus, OH 43210-1278

Phone 614-292-2651
FAX 614-292-9379
E-mail engosu@osu.edu

Date: 21 May 2009

To: Randy Smith
Vice Provost Academic Affairs

From: Ed McCaul
Secretary College Committee on Academic Affairs

Subject: Proposal for a Minor in Computational Science

Attached is a proposal from the Department of Computer Science and Engineering requesting approval of an undergraduate minor in Computational Science. The proposal was unanimously approved by the College of Engineering's Committee on Academic Affairs on the 18th of May 2009. If you have any questions about this proposal please contact P. Sadayappan or me.

Minor Program in Computational Science

Introduction

Computational science describes the application of computing, especially high performance computing, to the solution of scientific and technical problems. Computational scientists use computers to create mathematical models that help them simulate and understand the operation of natural and mechanical processes, as well as to visualize the operation and results of these models.

Computational science (i.e., science in-silico) has become a third way of advancing knowledge along with the traditional methods of theory and experimentation. In-silico simulations and modeling afford the opportunity to "see" the unattainable – phenomena that are too small (atoms and molecules), too large (galaxies and the universe), too fast (photosynthesis), too slow (geological processes), too complex (automobile engines), or too dangerous (toxic materials). In recent years, computational studies have produced enormous advances in almost all fields of scientific and engineering inquiry, including DNA sequencing, behavioral modeling, global climatic predictions, drug design, financial systems, and medical visualization.

In recognition of the importance of computational science, the Ohio Board of Regents created the Ralph Regula School of Computational Science (RRSCS) in December 2005. RRSCS is a "virtual" school housed at the Ohio Supercomputer Center that will not offer degrees of its own but will serve to organize and coordinate statewide efforts to integrate computational science programs into the curricula at participating institutions. This effort recognizes that the field is an interdisciplinary field with expertise scattered among different departments and different institutions. The intention of the school is to sponsor shared, inter-institutional programs that take advantage of the existing expertise, make it widely available, and limit the duplication of effort and expense where possible.

The first such program is a minor program in computational science intended initially for majors in science and engineering programs at the participating institutions. The program is being initiated as a part of a National Science Foundation grant to nine Ohio higher education institutions, the Ohio Learning Network, and the Ohio Supercomputer Center that pledged to cooperate in the creation of the program. The participating institutions are shown in Table 1. **Each of these institutions is separately approving the minor program so that it is officially established as a part of their degree programs. This is the proposal to do so at OSU.**

A minor is the most appropriate approach to this area at the undergraduate level because domain expertise extending from a field of science, mathematics, or engineering is required to understand and implement computational modeling. The minor program also will require fewer resources to implement while helping to organize the required inter-institutional agreements and operating procedures for the program.

Table 1: Institutions Participating in the NSF CI-TEAM Grant for Computational Science
Capital University
Central State University
Columbus State Community College
Kent State University
Ohio Learning Network
Ohio Supercomputer Center
Sinclair Community College
The Ohio State University
University of Cincinnati
Wittenberg University
Wright State University

Overview of the Minor Curriculum

The minor curriculum was created through a collaborative effort of the faculty at the participating institutions. Because of its inter-institutional nature, the curriculum is competency-based, defining learning outcomes as the basis for courses to allow for easier evaluation and approval of proposed courses. The competencies were also reviewed by a business advisory committee including major, prospective employers of our graduates. They helped to revise the proposed structure to best suit the need for graduates in the current market. The full description of the competencies is provided in Appendix A.

The proposed minor curriculum consists of six required courses and at least one elective. These are shown in Table 2. All students are required to take a year of calculus, which is part of the current requirements for the target major fields. Calculus could be taken concurrently with the introductory modeling and simulation course, but is a prerequisite for the other courses. The courses are described in the table of competencies shown in Appendix A.

In addition to the approval of the proposed minor program in Computational Science through the formal process at The Ohio State University, the courses proposed as a part of the curriculum will be reviewed by an inter-disciplinary, inter-institutional committee of computational science faculty from around the state to ensure that they meet the required competencies. The creation of course materials and conversion of existing courses to match the competencies and offer some courses at distance is being subsidized by an NSF grant and other funding at the RRSCS. The materials produced for those courses will be retained in a repository at RRSCS so that faculty can draw upon those shared materials to expand the number of courses offerings.

Coursework Requirements

The Computational Science minor requires the completion of at least 20 credits of approved coursework, including a required course in each of six designated areas and one elective, as summarized in Table 2.

Because of the inter-institutional nature of the program, students have two options for the minor courses. They can take the courses at OSU, if they are offered here, or they can take the courses at distance from one of the other participating institutions. Transfer credits from courses at other institutions will be handled in the usual way by the registrar's office under the collaborative agreement (details may be found in Appendix A, page 30). Table 3 shows a list of pertinent courses for the 2008-2009 academic year offered at different institutions. Thus OSU is not obligated to teach every course in the program; also some of the courses could be taught less frequently. In the current proposal, OSU courses are identified (Table 2) that meet or exceed the minimum competencies associated with the topics in the minor program. For some of the upper division courses that exceed the competencies, other pre-requisites are required (listed in Appendix B).

Required Courses

1) Simulation and Modeling: An introductory course on the use of models (continuous and discrete) and simulation in science and engineering. The overall goal is to introduce the need for modeling and simulation as an integral part of science and engineering practice, provide an introduction to modeling concepts, review the mathematics underlying deterministic models, build an understanding of both the computational and estimation errors associated with models, introduce stochastic or Monte Carlo simulations, and demonstrate an ability to formulate, use, and explain a project model. Greater detail on the list of competencies is available in Appendix A (page 14-16). A general introductory course in simulation and modeling is currently not offered at OSU. Students may take this course remotely at another institution (offerings for 2008-2009 are listed in Table 3; there are no prerequisites for the distance courses at other institutions as the level of expertise and mathematics required can be met by most college freshman upon entry), or use a more advanced domain-specific course that focuses on modeling and simulation. Table 2 lists suitable courses at OSU; details on pre-requisites and quarters of offering are provided in Appendix B.

2) Programming and Algorithms: A first course on computational thinking, use of a programming language for problem solving, and development of algorithms. In addition to several existing courses at OSU that would satisfy this requirement, a new Matlab-based course on computational thinking in the context of science and engineering (CSE 294P) is being developed by the CSE Department (pilot in Spring 2009), with science/engineering students as the target audience.

3) Numerical Methods: An applied introduction to use of numerical methods in solving linear and nonlinear equations, interpolation, numerical solution of differential equations. Because many engineering students either take a linear algebra course or are introduced to linear algebra concepts in numerical methods and the modeling and simulation course, we do not require a linear algebra course.

4) Optimization: Courses on the topic of optimization are being developed at some other institutions in support of a Computational Science minor. A course is also being considered for development by the IWSE department at OSU. Several departments have domain-specialized courses that focus on discrete and/or continuous optimization techniques, which may also be used to satisfy this requirement

5) Capstone Research/ Internship Experience: Each student must complete a guided research project or internship on a computational topic. The mechanism used to satisfy this requirement may differ across departments – e.g. computationally oriented senior design project or honors thesis, or a computationally oriented independent research study with a faculty member at Ohio State. The Ralph Regula School also plans to create a list of computationally oriented projects contributed from faculty at the participating universities; such a guided project can also be used to satisfy the capstone requirement. At OSU, the CSE Department will maintain a list of computational projects from faculty at Ohio State, and coordinate the matching of interested students with faculty guiding the projects. A student from any department may register for CSE 699 if they choose one of these computational projects for satisfying the computational capstone requirement.

6) Discipline-Specific Course: Any approved computationally oriented course from the student's major discipline. The current list of approved courses is listed in Table 2.

Elective Courses

At least one elective course must be completed, chosen from any of the following topics.

a) Differential Equations and Dynamical Systems: A course on numerical solution of differential equations, which are fundamental in modeling physical and biological systems. Many engineering majors are required to take differential equations as are those in some of the physical sciences. We did not make this a requirement in the minor program to accommodate majors in the biological sciences and fields where other mathematical or analytical approaches are more prevalent.

b) Parallel Computing: An introduction to parallel programming, parallel algorithm development, implementation and optimization. Courses offered at distance in this area generally require an introductory computer algorithms course or related programming course as a prerequisite.

c) Scientific Visualization: An introduction to tools and techniques for visualization of large-scale data produced by computational simulations. A version of this course will be offered by OSU CSE in Spring 2009.

Administration and Advising

The minor in Computational Science will be open to any undergraduate student at the Ohio State University. The Computer Science and Engineering department will serve as a primary agent for the proposed program, but a Computational Science Advisory Committee will be formed, with broader membership. The Committee will be chaired by a faculty member from CSE (appointed by the Department Chair), and it will include an additional faculty member from the College of Engineering (appointed by the Associate Dean of Undergraduate Education and Student Affairs), one faculty member from the College of Mathematical and Physical Sciences (appointed by the Associate Dean), and the Director of the Ralph Regula School of Computational Science. The Committee will meet at least once every year to review the minor curriculum and course offerings, and make revisions as necessary, including approval of new discipline-specific computationally oriented courses.

The undergraduate counseling office of the Department of Computer Science will assist with student advising for the Computational Science minor. It will make available a form that students may use to declare the Computational Science minor. Any petitions for substitution of alternative courses (taken at Ohio State or another university) to satisfy the requirements of the minor must be approved by the Computational Science Advisory Committee.

Because this minor program is unique and generally outside the required CSE curriculum, CSE majors will be allowed to take it.

For the capstone course, we expect the committee to create a set of general guidelines that will be given to the faculty with potentially eligible courses. The guidelines will ensure that approved courses or internships have sufficient computational applications to qualify as a

capstone for this minor program. Faculty will be asked to sign a form indicating that a particular course meets the guidelines and to submit the final summary reports from the projects annually to the minor program committee. This will allow the committee to evaluate whether the capstone courses are continuing to meet the guidelines as well as provide important information on the overall effectiveness of the program.

Table 2: Requirements for Undergraduate Computational Science Minor		
Topic	Courses	Required/Elective
Prerequisites		
Calculus	Math 151, 152, 153 (or equivalent, e.g. Math 161, 162 or Math H190, H191)	
Core Courses		
Simulation and Modeling	One of: CEG 640, CSE 778, ISE 521, 704, ME 785, MSE 533	Required
Programming and Algorithms	One of: CSE 202, CSE 294P, CSE 221, EG 167	Required
Numerical Methods	One of: AAE 581, CEG 406, CSE 541, ECE 715, Math 606, Math 607, ME 250	Required
Optimization	CEG 776, ECE 759, ISE 522, ME 761, MSE 600	Required
Discipline Specific Courses		
Capstone Research/Internship Experience	CE 660, CSE 699 or H783, ME 564 or 565, MSE 695 or other approved individualized research credits	Required
Discipline-specific Computationally oriented Course	CE/ECE 675, CSE 630, 655, 660, 670, 675, 680, Chem 644, MSE 756, Phys 780	Required
Elective: Choose at least one course from the following		
Differential Equations and Discrete Dynamical Systems	Math 255, Math 415, Math 568, Math 571	Elective
Parallel Programming	CSE 621	Elective
Scientific Visualization	CSE 694L	Elective

Table 3: 2008-2009 Academic Year Courses Offered at Collaborating Institutions			
Course Number and Title	Institution / Instructor	Quarter or Semester*	Credit Hours
PHY 212 Introduction to Modeling and Simulation	Sinclair Community College / Art Ross & Bob Chaney	Winter quarter 2009	4
PHY 220 Introduction to Computational Physics	Sinclair Community College / Art Ross	Spring quarter 2009	5
CSAC 245 Introduction to Computational Science	Capital University / Dr. Patrick Shields	Fall semester 2008	3
BSCI-50/70195 BSCI 40195/ BTEC-40220 ST: Bioinformatics	Kent State University / Dr. Helen Piontkivska	Fall semester 2008	3
20-CS-668 Parallel Computing	University of Cincinnati / Dr. Fred Annexstein	Fall Quarter 2008	3
Chem 644 Computational Chemistry	The Ohio State University / Dr. Richard Spinney	Fall Quarter 2008 Classroom only	3
CSE 694L Data and Information Visualization Description	The Ohio State University / Dr. Raghu Machiraju	Spring Quarter 2009	4
COMP 345 Optimization	Wittenberg University / Eric A. Stahlberg	Fall Semester 2008 Classroom only	3
MATH 299 Special Topics: Differential Equations and Discrete Dynamical Systems	Columbus State Community College / John Nedel	Spring quarter 2009 Classroom only	6
CS/PBIO 416/516 Problem Solving with Bioinformatics Tools	Ohio University / Dr. Lonnie Welch and Sarah Wyatt	Spring Quarter 2009	4
CS 412 Parallel Computing	Ohio University / Dr. Frank Drews	Spring Quarter 2009	5
CST120 Computational Science Methods	Stark State Community College / Robert Berens	Fall Semester 2008 Classroom only	3
CST121 Introduction to Modeling and Simulation	Stark State Community College / Karen Hardesty	Spring Semester 2009 Classroom only	3
CSA 443 High Performance Computing and Parallel Programming	Miami University / Dr. Dhananjai M. Rao	Spring Semester 2009	3
15PHYS410 Computational Physics	University of Cincinnati / Richard Gass	Spring Quarter 2009	3

*All courses offered at distance unless otherwise indicated.

Some of the courses will be given temporary course numbers for the first year as they go through the approval process for a permanent course number at the participating institutions.

The RRSCS will maintain a central website that students and faculty can use to see what courses and being offered. The site will also show how to register for courses being offered at other

institutions. The proposed mechanisms for these procedures are currently being discussed and are discussed briefly below. The proposed inter-institutional agreements have been discussed at all of the participating institutions and we are working with the appropriate offices to implement those agreements. The agreement is shown as Appendix A.

Inter-institutional Arrangements

As part of the NSF project, a subcommittee of representatives of academic affairs officers and registrars has been meeting to discuss the inter-institutional arrangements for the minor program. They have agreed in principle to several things:

- The registration process for students should be as easy as possible.
- RRSCS should maintain a list of approved courses by year so it is relatively easy to conduct degree audits.
- Transfer of credits from other institutions can be handled by the registrars at the institutions as those courses are completed.
- Tuition will be collected at the home institution of the student based on the total number of credit hours taken including hours taken from other institutions and charged at the home institution rate.
- There should be some compensation given to the institutions teaching courses to students at other institutions in lieu of tuition.
- Petitions for exceptions to the required curriculum and other individual advising will remain solely at the home institution.
- Mechanisms for changes and additions to the minor curriculum should be undertaken by the RRSCS and a statewide committee of faculty.

Appendix A
Computational Science Minor Program
Detailed Program Description

**Ralph Regula School of Computational Science
Cyberinfrastructure Team Project
Summary of Undergraduate Minor Program Requirements
September 2006**

The minor in computational science will be offered by a number of higher education institutions in Ohio. Degrees will be offered by the students "home" institution but it will be possible for the student to take minor program materials from other, participating institutions. We have started with a minor because we believe that each student needs some domain expertise in a major field before being able to complete computationally-based projects in related areas.

There is a committee reviewing the various options for the sharing of materials, instruction, and revenues associated with cross-registered students. They will make some recommendations to the participating institutions in the coming months on the institutional procedures.

Because of the inter-institutional nature of the program, we have designed it to be competency-based. The instructional materials associated with each competency or subset of competencies can then be embedded either in existing courses on each campus, in new courses at one or more campuses, or as stand-alone modules that might be taken at distance and at the time that each student is ready for them. Competency will then be demonstrated through one or more assessments of the student's abilities on a combination of exams and projects. The competency-based approach is preferred because it gives curriculum flexibility to participating programs and gives employers some assurance of the working knowledge of each graduate.

The draft competencies created by the participating faculty has been reviewed by a business advisory committee. They confirmed the majority of the recommended competencies at a meeting on May 31, 2006 and offered some advice on topic emphasis and breadth. This document reflects their comments and is the basis on which faculty are currently proposing instructional modules that meet the competencies.

The computational science minor will enable science and engineering majors to apply computational tools to the problems in their discipline. As such, the minor breaks into three broad categories: Prerequisites, the Computational Science Core Competencies and the Discipline Specific Competencies. At this stage, we are focused on the competencies that comprise the Computational Science Core. The goal of the Core competencies is to establish a foundation that can be leveraged in the Discipline Specific Competencies that are directly applicable to the practice of the student's chosen profession. The competencies are shown in Table 1.

Our remaining decisions involve deciding what competencies are associated with each broad area, how many elective competencies are required for the minor, and the specific competencies for each discipline-oriented course. Those will be addressed in Autumn 2006 as we begin to pre-test the materials in classrooms around Ohio.

Table 1: Competencies and Requirements for Undergraduate Computational Science Minor		
Topic	Subtopics	Required/Optional
Prerequisites		
Calculus 1 and 2		Required
Computational Science Courses		
Simulation and Modeling	Should use one of the major symbolic programs Maple, MATLAB, Mathematica; need for local, hands-on support at outset of course	Required
Programming and Algorithms	Logic of programming whether a traditional computer science or programming with a symbolic language like Maple, MATLAB, Mathematica	Required
Differential Equations and Discrete Dynamical Systems	Depending upon major; linear algebra may also be needed by some	Elective
Numerical Methods	Need for a project-based course which touches the most important topics rather than a standard math course	Required
Optimization	An important topic; could be part of a modeling course or integrated with numerical methods	Required
Parallel Programming		Elective
Scientific Visualization		Elective
Discipline Specific Courses		
Capstone Research/Internship Experience		Required
Discipline Oriented Courses		One required; probably only one per field for awhile

**Minor Program in Computational Science
Competency/Topic Overview
Area 1: Simulation and Modeling**

Competency/Descriptors
<p>Explain the role of modeling in science and engineering Descriptors: Discuss the importance of modeling to science and engineering Discuss the history and need for modeling Discuss the cost effectiveness of modeling Discuss the time-effect of modeling (e.g. the ability to predict the weather) Define the terms associated with modeling to science and engineering List questions that would check/validate model results Describe future trends and issues in science and engineering Identify specific industry related examples of modeling in engineering (e.g., Battelle; P&G, material science, manufacturing, bioscience, etc.) Discuss application across various industries (e.g., economics, health, etc.)</p>
<p>Analyze modeling and simulation in computational science Descriptors: Identify different types of models and simulations Describe a model in terms of iterative process, linking physical and virtual worlds and the science of prediction Explain the use of models and simulation in hypothesis testing (e.g. scientific method)</p>
<p>Create a conceptual model Descriptors: Illustrate a conceptual modeling process through examples Identify the key parameters of the model Estimate model outcomes Utilize modeling software and/or spreadsheets to implement model algebraic equations (e.g. Vensim, Excel, MATLAB, Mathematica) Construct a simple computer visualization of the model results (e.g. infectious disease model, traffic flow, etc.) Validate the model with data Discuss model quality and the sources of errors</p>
<p>Examine various mathematical representations of functions Descriptors: Describe linear functions Define non-linear functions (e.g., polynomials, exponential, periodic, parameterized, etc.) Visualize functions utilizing software (e.g. Excel, Function flyer, etc.) Determine appropriate functional form to fit the data Demonstrate essential mathematical concepts related to modeling and simulation</p>
<p>Analyze issues in accuracy and precision Descriptors: Describe various types of numerical and experimental errors Explain the concept of systematic errors</p>

<p>Explain the concept of data dependent errors Illustrate calculation and measurement accuracy Identify sources of errors in modeling and approaches to checking whether model results are reasonable</p>
<p>Understand discrete and difference-based computer models Descriptors: Explain the transition of a continuous function to its discrete computer representation Represent “rate of change” using finite differences Cite examples of finite differences Explain derivatives and how they relate to model implementation on a computer Write pseudo-code for finite difference modeling</p>
<p>Demonstrate computational programming utilizing a higher level language or modeling tool (e.g. Maple, MATLABTM, Mathematica, other) Descriptors: Describe the system syntax (e.g., menus, toolbars, etc.) Define elementary representations, functions, matrices – arrays, script files, etc. Explain programming and scripting processes (e.g., relational operations, logical operations, condition statements, loops, debugging programs, etc.) Create tabular and visual outputs (e.g., 2-D and 3-D subplots) Translate the conceptual models to run with this system and assess the model results (e.g. traffic flow and/or “spread of infectious disease”) Illustrate other people’s models utilizing the modeling program</p>
<p>Assess computational models Descriptors: Assess problems with algorithms and computer accuracy Discuss techniques and standards for reviewing models Verify and validate the model Discuss the differences between the predicted outcomes of the model and the computed outcomes and relevance to the problem Discuss the suitability and limits of the model to address the problem for which the model was designed</p>
<p>Build event-based models Descriptors: Describe event-based modeling (e.g. SIMULINKTM; Extend, ARENA) Run existing models Translate conceptual models (e.g., traffic flow utilizing SIMULINKTM)</p>
<p>Complete a team-based, real-world model project Descriptors: Identify a problem, create mathematical model and translate to computational modeling Organize and present project proposal Document model development and implementation Collaborate with team members to complete the project</p>
<p>Demonstrate technical communication Descriptors: Demonstrate technical writing skills in the comprehensive report</p>

Demonstrate verbal communication skills in an oral presentation
Create and present visual representation of model and results
Address all components of a comprehensive technical report
Respond to peer review

**Minor Program in Computational Science
Competency/Topic Overview
Area 2: Programming and Algorithms**

Competency/Descriptors
<p>Describe the fundamentals of problem solving</p> <p>Descriptors:</p> <p>Understand Top-Down thinking and program design Discuss breaking up a problem into its component tasks Understand how tasks acquire data Describe how tasks should be ordered Represent tasks in a flow-chart style format Understand the difference between high-level languages (for example Mathematica, Maple or MATLAB), medium level languages (for example FORTRAN or C) and low-level languages (assembler) and when each should be used.</p>
<p>Understand and write Pseudo code</p> <p>Descriptors:</p> <p>List the basic programming elements of Pseudo code Explain the logic behind an if/then/else statement Understand the iterative behavior of loops Describe the difference between several looping constructs Write Pseudo code to solve basic problems Understand how to represent data flow in and out of subprograms.</p>
<p>Use subprograms in program design</p> <p>Descriptors:</p> <p>Describe how logical tasks can be implemented as subprograms Understand the logical distinction between functions and subroutines Explain the control flow when a function is called Define dummy and actual arguments Discuss the different relationships dummy and actual arguments Explain how function output is used Understand how languages handle passed data into functions and subprograms, especially one and two dimensional arrays.</p>
<p>Write code in a Programming language</p> <p>Descriptors:</p> <p>Understand the concept of syntax in a programming language Describe the syntax of the programming language constructs List the type of subprograms available in the language Explain the concepts of argument pass-by-value and pass-by-reference Understand what a compiler and linker do Understand the difference between a compiled and interpreted language Understand the difference between a typed and an un-typed language Understand the difference between a source file and an executable file Write and run basic programs in the language of choice Understand how to de-bug code and how to "sanity check" code. Understand the importance of user-interfaces: clear input instructions including physical</p>

<p>units if needed and clearly formatted and labeled output</p> <p>Understand the numerical limits of various data types and the implications for numerical accuracy of results.</p>
<p>Use different approaches to data I/O in a program</p> <p>Descriptors:</p> <p>Explain the advantages and disadvantages of file I/O</p> <p>Describe the syntax for file I/O in your programming language</p> <p>Compare binary and ASCII file I/O</p> <p>Write code using file I/O and keyboard/monitor I/O</p>
<p>Understanding and use of fundamental programming Algorithms</p> <p>Descriptors:</p> <p>Explain an algorithm as an ordered series of solution steps</p> <p>Describe an algorithm for a simple programming problem</p> <p>Learn and use “classic” programming algorithms from a field of interest to the student. If possible, these should be algorithms used in the student’s discipline.</p> <p>Describe what a software library is</p> <p>Understand how library functions implement algorithms</p> <p>Write code to implement your own version of “classic” algorithm</p> <p>Compare with code using a library function</p> <p>Understand data flow into library functions and implications of selecting any “tuning parameters” or options that may be required.</p>
<p>Explain various approaches to Program Design</p> <p>Descriptors:</p> <p>Describe Functional decomposition (Top-down Problem Solving)</p> <p>Be familiar with different programming styles (e.g. function, procedural, rule based)</p> <p>Understand how to modularize code</p> <p>Understand the benefits of code re-use</p> <p>Explain the operation of a Boss-Worker design</p> <p>Compare designs based on Global Variables vs. self-contained functions</p> <p>Define Object-Oriented Programming (OOP)</p> <p>Contrast OOP with functional decomposition</p> <p>Explain the power of Inheritance in OOP</p> <p>Understand how to document code</p> <p>Understand how to write and when to use stubs and drivers.</p>

**Minor Program in Computational Science
Competency/Topic Overview
Area 3: Differential Equations and Discrete Dynamical Systems**

Competency/Descriptors
<p>Describe the solution methodology for first order linear differential and difference equations</p> <p>Descriptors: Analyze modeling problems with first order differential equations and present their solution methodology (e.g. liner, homogeneous, exact) Analyze modeling problems with first order difference equations and present their solution methodology (e.g. homogeneous, non-homogeneous). Analyze long term behavior</p>
<p>Describe the solution methodology for systems of linear first order differential and difference equations</p> <p>Descriptors: Describe modeling problems with systems of first order differential equations and present their solution methodology (e.g., homogeneous with constant coefficients, variation of parameters) Describe modeling problems with systems of first order difference equations and their solution methodology (e.g., homogeneous with constant coefficients)</p>
<p>Describe the solution methodology for higher order differential and difference equations</p> <p>Descriptors: Describe modeling problems with higher order differential equations analyze their solution methodology (e.g., homogeneous, non-homogeneous, undetermined coefficients, variation of parameters) Describe modeling problems with higher order difference equations analyze their solution methodology (e.g., homogeneous, non-homogeneous). Analyze the long-term behavior.</p>
<p>Describe the solution methodology for differential equations using the Laplace Transforms</p> <p>Descriptors: Discuss the Laplace transformation of (e.g., continuous , discontinuous, delta and convolution) functions Describe modeling problems with differential equations and present their solution methodology using Laplace transformations (use of CAS, Maple, Mathematica)</p>
<p>Describe the solution methodology for non-linear differential equations</p> <p>Descriptors: Describe the concept of an equilibrium point Model with non-linear differential equations and present the phase –portrait analysis Understand and demonstrate how chaos is generated in the solution process of non-</p>

linear differential equations.

Describe the solution methodology for non-linear difference equations

Descriptors:

Describe the method of linearization

Describe the concepts of Logistic and Henon Maps

Model with non-linear difference equations and demonstrate understanding of fundamental concepts from Bifurcation theory (e.g., fixed, periodic points, chaos)

Describe techniques for controlling chaos

Understand concepts of numerical accuracy applied to each solution approach

**Minor Program in Computational Science
Competency/Topic Overview
Area 4: Numerical Methods**

Competency/Descriptors
<p>Understand number representation and computer errors Descriptors: Understand the pros and cons of floating point and integer arithmetic Describe various kinds of computing errors (e.g., round-off, chopping) Describe absolute, relative error and percent error Discuss error propagation Describe loss of significance – methods to avoid loss of significance</p>
<p>Analyze methods for solving non-linear equations Descriptors: Discuss and contrast fixed point methods (e.g., bisection, secant, Newton’s) for a single equation Describe a fixed point method for a system of equations (e.g., Newton’s)</p>
<p>Describe techniques for solving systems of linear equations Descriptors: Describe the naïve Gauss elimination and the partial pivoting method Understand the concepts of condition number and ill-conditioning problems Discuss and contrast factorization methods (e.g., LU, QR, Cholesky, SVD) Discuss and contrast iterative methods (e.g., Jacobi, Gauss Siedel) Describe convergence and stopping criteria of iterative methods</p>
<p>Analyze techniques for computing eigenvalues—eigenvectors (Optional) Descriptors: Describe and give examples of eigenvalue –eigenvector problems using specific, applied examples and their significance Describe canonical forms of matrices Describe and contrast direct methods for computing eigenvalues (e.g., power method, inverse power method) Describe and contrast transformation methods (e.g., QR algorithm)</p>
<p>Describe interpolation and approximation methods Descriptors: Describe and contrast interpolation methods (e.g., Lagrange, Chebyshev, FFT) Describe interpolation with spline functions (e.g., piecewise linear, quadratic, natural cubic) Discuss approximation using the method of least squares (linear .vs. non-linear)</p>
<p>Describe numerical methods for Ordinary Differential Equations Descriptors: Describe and compare basic methods for IVPs (e.g., Euler, Taylor, Runge-Kutta) Describe and compare predictor-corrector methods Describe and compare multistep methods Discuss and contrast numerical methods for BVPs (e.g., shooting method, finite difference method) Compare the accuracy, memory requirements, and precision of each of the approaches</p>

Describe numerical methods for Partial Differential Equations

Descriptors:

Describe and compare numerical methods for parabolic PDEs (e.g., finite difference, Crank-Nicolson)

Describe numerical methods for hyperbolic PDEs

Describe numerical methods for elliptic PDEs (e.g. finite difference, Gauss-Seidel)

Discuss the finite element method for solving PDEs

Describe Monte Carlo Methods

Describe applications of Monte Carlo models with examples

Discuss algorithms for Monte Carlo methods

**Minor Program in Computational Science
Competency/Topic Overview
Area 5: Optimization**

Describe and use Optimization techniques

Descriptors:

Describe and contrast unconstrained optimization methods (e.g., Golden section search, Steepest descent, Newton’s method, conjugate gradient, simulated annealing, genetic algorithms)

Describe and contrast constrained optimization methods (e.g., Lagrange multiplier, quasi-Newton, penalty function method)

Implement linear and non-linear programs

Analyze linear programming methods (e.g., simplex method)

Describe non-linear programming methods (e.g., interior, exterior, mixed methods)

Demonstrate ability to correctly use software systems (e.g., Matlab, IMSL, NAG) to solve practical optimization problems

**Minor Program in Computational Science
Competency/Topic Overview
Area 6: Parallel Programming**

Competency/Descriptors
<p>Describe the fundamental concepts of parallel programming and related architectures</p> <p>Descriptors: Describe the differences between distributed and shared memory architectures Describe the difference between domain and functional decomposition in parallel Describe a parallel programming approach to an introductory problem Compare parallel, distributed, and grid computing concepts</p>
<p>Demonstrate parallel programming concepts using MPI</p> <p>Descriptors: Describe the MPI programming model Create, compile, and run an MPI parallel program Create MPI programs that utilize point-to-point communications Create an MPI program that uses point-to-point blocking communications Create an MPI program that uses point-to-point non-blocking communications Create an MPI program that uses collective communications Create an MPI programs that use parallel I/O Create MPI programs that use derived data types Create MPI programs that use vector derived data type Create MPI programs that use structure derived data type</p>
<p>Demonstrate knowledge of parallel scalability</p> <p>Descriptors: Use mathematical formulas to determine speed-up and efficiency metrics for a parallel algorithm. Demonstrate the use of graphical systems such as MATLAB to display speed-up and efficiency graphs</p>
<p>Demonstrate knowledge of parallel programming libraries and tools</p> <p>Descriptors: Demonstrate the use of performance tools for profiling programs (e.g., GNU GPROF or MATLAB profiler) Create parallel programs with calls to parallel libraries (e.g. BLAS, BLACS, ScaLAPACK or FFTW) Demonstrate the use of MPI tracing tools (e.g., VAMPIR) to determine parallel performance bottlenecks</p>

**Minor Program in Computational Science
Competency/Topic Overview
Area 7: Scientific Visualization**

<p>Competency/Descriptors</p> <p>Define SciVis needs; relationships to human visualization; basic techniques Define Scientific Visualization (Sci Vis) Discuss needs of SciVis (in the framework of a large variety of possible application areas) Survey different platforms for Visualization (e.g. AVS, VTK, OpenGL, VRLM) Discuss the different techniques and visualization methods used in SciVis Explain the human visualization system – capabilities and perceptions Explain the different steps in the visualization pipeline Discuss different sources of data for SciVis and explain the terms applied to data types (i.e. scalar, vector, normal, tensor) Discuss different types of grids (e.g., regular vs. irregular grids) Discuss the different methods used to gather data Describe and explore the use of different file formats for sharing data (netCDF, XML, TIFF, GIF, JPEG, Wavefront OBJ) Discuss limitations of different methods Discuss future applications in emerging fields Metadata needs for graphics libraries</p>
<p>Overview of computer graphic concepts Descriptors: Overview of SciVis concepts (pixels, rgb colors, 3D coordinate system, mapping 3D data to a 2Dscreen, continuous vs. discrete) Discuss polygonal representation Discuss lighting/shading Overview of classification/segmentation and transfer functions Discuss concept of rendering pipeline (no details about matrices) Discuss hardware rendering (mainly for polygonal models, few specialized volumetric hardware cards) Identify terms used in virtual space and in graphics elements Navigate in virtual space and manipulate primitive objects</p> <ul style="list-style-type: none"> ▪ Transform: scale, rotate, translate) ▪ Manipulate surface ▪ Manipulate lighting and camera <p>Explore colormaps and examine conceptual definitions for different color maps (pertaining to color spaces HSV, RGB, etc.) as related to representing data and relationships to perception</p>
<p>Describe approaches to visualization for different scientific problems Descriptors: Examine different computational solutions to scientific problems Explain the different techniques used in visualization (i.e. glyphs, iso-contours, streamlines, image processing, volume-data) Examine the application of problems to visualization techniques</p>

Utilize software tools to implement visual image of a solution
Discuss the use of time in animation

Utilize software to implement grid representations of data

Descriptors:

Identify the various cell representations (i.e. points, polygons, 3d geometries)

Discuss the application to different grid types (i.e. structured, unstructured, random)

Discuss raycasting methods and texture mapping

Examine the details of raycasting sampling (FAT(low resolution sampling), interpolation techniques).

Examine algorithms: Direct Composite, SFP, use of transparency.

Identify the grid representation and data(color reps.) (regular grids, 1, 8, 24 and 32 bit color information)

Discuss algorithms for manipulating images, distortion, fft's, enhancement, restoration, frequency domains

Utilize software to implement different grid types

Discuss limitations of grids

<p>Use visualization software to display an isosurface</p> <p>Descriptors: Discuss different data types used: scalar vs. vector data Discuss the different grid types Discuss the different algorithms (Marching Cubes etc) Introduce details of the system being used in a course (e.g., VTK, AVS, etc.) Apply the system to extract and display an isosurface of some data set (could be tailored towards the teacher's and student's interests/application areas) Discuss limitations of these methods</p>
<p>Use visualization software to complete a volumetric rendering</p> <p>Descriptors: Discuss direct volumetric rendering (raycasting and texture mapping) and its advantages/disadvantages vs. surface rendering Discuss segmentation/classification and transfer functions Discuss and illustrate how to use a system (VTK, AVS, etc.) to do volumetric rendering Using the system, visualize a data set using raycasting Using the system, visualize a data set using texture mapping Discuss limitations of this method</p>
<p>Utilize visualization software to visualize a vector dataset</p> <p>Descriptors: Discuss vector data Discuss different methods for vector visualizing (particles, stream ribbons, vector glyphs, etc.) Discuss the use of structured grid types: (ir)regular, cylindrical, spherical Discuss application areas for vector visualization (air flow, etc.) Using a system (VTK, AVS, etc.), visualize a vector data set Discuss limitations of this method</p>
<p>Explore examples of image processing</p> <p>Descriptors: Discuss basic steps and goals in image processing Discuss variety of data sources of images and how they can be represented Discuss algorithms used for image processing Explore examples of image processing (e.g., noise reduction, image enhancement, feature extraction etc) Discuss challenges and limitations in image processing</p>
<p>Use advanced techniques applied to a real problem</p> <p>Descriptors: To be chosen by instructor based on instructor/student interest. Among suggested topics are:</p> <ul style="list-style-type: none"> - visualizing an irregular grid; - visualizing a data set specific to the area of interest (see 3.2 and 3.3 for specific examples) - writing a segmentation tool - implementing a visualization algorithm from scratch (such as marching cubes or raycasting)
<p>Examine SciVis problems for Biological Sciences – "OMICS" applications</p>

Descriptors:

Examine different problems existing in 'OMICS sciences that require visualization solution (overview)

Discuss challenges of representing biomedical/biological data (i.e., representing protein structure or genomic sequence with all their attributes as a visual metaphor)

Discuss challenges associated with visualization of scattered data such as text information and bioinformatics data (e.g., phylogenetic information)

Gene finding in genomic sequences

- Examine different components of a gene structure
- Visualize genomic structure of an individual gene
- Build a comparison between genomic features from several genomes
 - o Visualize (and examine) similarities and differences
 - o Discuss goal-dependent options of parsing the results to be explored elsewhere (e.g., as plain text, XML-marked)

Protein folding and protein structure prediction

- o Discuss the differences between protein folding and protein structure prediction
- o Explore different methods used in protein folding and structure prediction
- o Apply different methods of protein structure prediction and compare the results
- o Construct, visualize and examine structure-based protein alignment

Biological networks (e.g., protein-protein or protein-DNA interaction networks)

Visualization of various types of expression data

- o Discuss and contrast different types of expression data – e.g., microarray gene expression data, protein expression data
- o Discuss different visualization (and analyses) techniques used for expression data
- o Apply (and compare outcomes) hierarchical clustering and k-means clustering to the same gene microarray expression data
- o Discuss pros and cons of different clustering methods, their shortcomings, and ways to access the quality of clusters

Discuss potential applications of SciVis techniques in biomedical and drug design fields

Utilize MATLAB to implement/solve the above problems

Explore SciVis techniques in BioMedical applications**Descriptors:**

Explore a variety of biomedical applications of SciVis to explore large datasets such as MRI and confocal microscopy data

Overview of volume visualization techniques in biomedical problems

Examine and different ways MRI (Magnetic Resonance Imaging) data can be visualized (e.g., 2-D image versus isocontour slices).

Discuss potential applications (interpretation) of each of the techniques.

Utilize software tools (MATLAB, VTK) to apply the techniques above



RALPH REGULA SCHOOL OF COMPUTATIONAL SCIENCE CONSORTIAL AGREEMENT

REGULA SCHOOL CURRICULUM

Institutions participating in the Ralph Regula School of Computational Science will offer a multi-institutional, interdisciplinary undergraduate minor in computational science with courses starting in the fall of 2007. The effective date of this agreement will be August 15, 2007. Standardized certificate programs to create workforce knowledge and skills valued by industry also are under development. Graduate courses will be added by 2009. The undergraduate minor curriculum offering for 2007-2008 is as follows:

Course Title and Number	Institution/Instructor	Quarter or Semester*	Credit Hours
CS 399 Selected Topics in Computational Science Programming and Algorithms	Wright State Ronald Taylor, Computer Science	Fall quarter 2007	4
Special Topics Introduction to Modeling and Simulation Part 1	Sinclair Art Ross Physics	Fall quarter 2007	2
Special Topics Introduction to Modeling and Simulation Part 2	Sinclair Art Ross Physics	Winter quarter 2008	2
Computational physics	Art Ross Physics	Spring quarter 2006	4
CSAC 245 Introduction to Modeling and Simulation	Capital University John Philips and Pat Shields	Fall semester 2007	3
Bioinformatics	Kent State Helen Piontkivska	Fall semester 2007	3
Parallel and Distributed Computing	Univ. Cincinnati Fred Annexstein	Fall quarter 2007	3
Computational Physics 15PHYS410	Univ. Cincinnati Richard Gass	Spring 2008	3
Computational Thinking in Context CSE294 (Programming and Algorithms)	OSU, CSE P. Sadayyapan	Spring, 2008 Classroom only	4
Chem 644 Computational Chemistry	Chris Hadad OSU	Spring, 2008 Classroom only	3
Optimization	Wittenberg	Still pending	5
CPS3465 Parallel Programming	Robert Marcus Central State	Spring Semester 2008	3
Differential Equations Using Computation, MATH299	Columbus State	Summer quarter 2008	6

*All courses offered at distance unless otherwise indicated.

The full curriculum description can be found at:

<http://www.rrscs.org/docs/competencyfinal.pdf>

Regula School Initial Participating Institutions

Capital University
Central State University
Columbus State Community College
Kent State University
Ohio State University
Sinclair Community College
University of Cincinnati
Wittenberg University
Wright State University

Terms

- **Home Institution:** The institution where the student is admitted as a student. Home students “visit” other institutions through Regula School courses
- **Host Institution:** The institution where the Faculty of Record (primary course instructor) is employed. Host institutions “teach” the Regula course.
- **Participating Institutions:** Colleges and universities that elected to participate in sharing courses through the Regula School.
- **Faculty Coordinators:** At least one faculty member at each participating institution responsible for assisting students to register for courses, shepherding courses for approval by deans, department heads and curriculum committees, arranging course technical support, as needed, and communicating with their Home Registrar and the Regula School.
- **Registrar Contacts:** The staff in the Office of the Registrar of participating institutions who are responsible for sending and/or receiving Regula -related information to the Faculty Coordinators, the Host or Home Registrars Contacts, and the Regula School
- **Technical Contacts:** The staff at participating institutions responsible for providing various related technical support (video conferencing, student support services, etc.)
- **Visiting Students:** Students from institutions other than the Host Institution, who participate in Regula School courses.

REGULA SCHOOL START UP SUPPORT

To ensure that students have continual access to Regula School courses, the School will assist institutions with start-up costs. Each institution will provide the agreed upon course with the faculty-approved competencies for \$3000 one time for an offering of the course. The courses and contracted institutions are listed above.

REGISTRATION PROCESSES

Regula School course information and schedule is made available through the Regula School website and the websites and print catalogs of all participating institutions (<http://www.rrscs.org>).

Once a course has been approved through its local curriculum review process and/or designated for a Regula course, that information will be posted at the host institution on their website, on-line catalog, and course management system (if available) and shared with the Regula School. If faculty approval is required by a student interested in registering for the Regula course, the Faculty Coordinator at each institution will secure approval for the student.

Supplementary Regula School course information will be available for student viewing on the Regula School web site (<http://www.rrscs.org/minorcourses/index.shtml>)

The Regula School will maintain an informational web site for students to gain knowledge about the course and curriculum, internships, and job possibilities. The Regula School will link to each participating institution's policies related to class calendars and dates for adds, drops, withdrawals, grade translations. This web site will stress that students register for courses through their Home Institution and not through this web site. Faculty Coordinators will provide Regula course information to the Regula School for publication on the RRSCS web site.

Students register and pay tuition for Regula course(s) at their Home Institution. To provide the information required by the registrars to track cross-registering students, students will be asked to fill out a common form for each course enrollment as a visiting student. The procedure used will be:

1. When a student fills out the special common form (Application for Host Institution Class Enrollment attached to this agreement) registrars at each participating home institution will add a specially tagged course number or a placeholder class on their schedule to enroll them in the class. A long-term goal of the consortium will be to cross-list all Regula School courses in the regular catalogs to facilitate registration and grade transfer procedures. The RRSCS website will have information about the course calendar for each such course and students will need to comply with the calendar at the host institution. The home institution will inform the student that, in cases where the home and host institutions differ in academic calendar systems (i.e., semester vs. quarter), the term associated with the home institution's placeholder course may not correspond to the actual term associated with the course taught at the host institution.
2. Students will register for the course at their home institution using regular registration procedures.
3. In addition, students will fill out the Application for Host Institution Class Enrollment, sign it, and turn it in at their home institution registrar's office. When faculty approval is required for registration, Faculty Coordinators will obtain approval for registration. Based on the local enrollment systems, each Regula School course will reserve one half of the enrollments in each class for the visiting students up to 30 days prior to the start of the class and after that time, class enrollment should become open to all students. To assist this process, registrars and Faculty Coordinators will meet online once per term.
4. Once the Application for Host Institution Class Enrollment has been received, the home registrars will fax the completed form to the host registrar who will, in turn, add the student to their local database and register the student for the hosted course, and notify the home institution of

completion of the registration process. This process will be facilitated electronically when those capabilities are in place through the Ohio Board of Regents.

5. The host institution will then notify the student that they are registered and transmit the procedures for them to obtain access to course materials and other host institution systems. Students will be given the same privileges as local students. **As required, network IDs and Passwords are mailed to Regula students by the appropriate administrative office at the host university.**
6. Host/Teaching and Home Registrar Contacts will distribute course rosters. The Host/Teaching Registrar Contact will send the complete course roster (including visiting students and regular students) to the Faculty/Instructor of Record per usual systems. The Home Registrar Contact will send a course roster of only the visiting students from that institution to their local Faculty Coordinator. All participating Registrars will send an email to the Regula School that summarizes the institution's final student enrollment numbers (undergraduate and graduate) in the shared course. This email to the Regula School should NOT include specific student information, only numbers of undergraduate and graduate students enrolled and their home institutions
7. **Drops/adds/withdrawals will be managed and communicated by Host/Teaching and Home Registrar Contacts.** Students will agree to abide by the calendar of the teaching institution in term of dates for withdrawal, penalties, and final grades. This information will be posted on the RRSCS website and the teaching institution site. Home Institution procedures and policies are followed regarding any administrative fees that are charged for add/drop activity, late enrollment processing, etc. Withdrawals should be made through both the home and host institutions. As a secondary measure, the home institution also will communicate the student's withdrawal to the host institution's registrar. Mechanisms to track drops will need to put into place to allow this information exchange.

GRADES AND TRANSCRIPTS

Grading Policies

Grade conversions will happen at the Home Institution. The Faculty of Record will submit all student grades (visiting and regular) to the Host/Teaching Registrar Contact using the grading scale of the Host/Teaching Institution.

Upon completion of the academic term, the host institution will notify the home institution of the final grade in the course and number of credit hours completed. The home institution will be responsible for all grade conversions and course grading scale adjustments. As these are completed, the home institution will provide a list of students and the courses they completed.

Grade changes will follow Home Institution policies and procedures

Local policies and procedures of the student's Home Institution regarding late, missing, incorrect grades, and their impact on graduation will apply. In cases where graduations at Home Institutions are earlier than the Host/Teaching Institution, students will work with Faculty Coordinators to resolve any issues that might impact graduation.

STATE SHARE OF INSTRUCTION

State Share of Instruction will remain with the home institution.

Public colleges and universities offering Regula courses will report student enrollments through regular HEI reporting mechanisms. State Share of Instruction (SSI) will flow to the home institution by agreement. Independent non-for-profit institutions will not receive state support in the form of State Share of Instruction (SSI).

FUNDING MODELS

There are several aspects to the funding model under this collaborative agreement. To assist in the implementation of the program, the Ralph Regula School of Computational Science will offer a one-time subsidy of \$3000 per course for each of the courses taught in the 2007-2008 academic year.

Each institution will collect tuition from the students taking courses taken through the Regula School. There will be no differences in course funding with native students. The home institution will collect the tuition and fees and state subsidy for those students.

Revenue will be shared with the Regula School and the institutions hosting visiting students. Tuition will be collected at the home institution for those students using their current tuition and fee structure. Payment will be made to the Regula School at a rate of \$250 per quarter credit hour for each student hosted by other institutions. Public institutions will continue to collect the State Share of Instruction for those students.

The Regula School will use 25% of the collected fees for additional course development, marketing, and administration. The balance of the fees collected for hosted students will be passed to the hosting institution.

CURRICULUM APPROVAL AND INSTITUTIONAL INVOLVEMENT

All institutions participating in this consortium are expected to play an active role in on-going curriculum development, review, and delivery.

- A) Each participating institution will offer at least one Regula School class at the institution in each academic year.
- B) Each institution will appoint one faculty representative to the statewide curriculum review committee. That committee will meet quarterly (if needed) but at least twice per year to review and approve new courses for existing consortium members or proposals for courses by new member institutions. The committee also will review the competencies required in the minor program and revise them as necessary in response to changes in the science and engineering fields.
- C) Member institutions can propose changes to the curriculum or to the consortium agreement. The changes will be considered by the statewide committee and a recommendation made to the consortium for their approval.

CURRICULUM EVALUATION

Curriculum and course reviews are the hallmarks of quality processes in higher education. Due to the nature of the content, courses in the Ralph Regula School of Computational Science have a shorter 'shelf life' than some other academic disciplines. A continuous improvement process is necessary for the strongest curriculum and the best trained students. The process shall be as follows:

A) The curriculum for the minor in computational science will be reviewed at the end of the first full year of all courses taught. A review team of faculty, students, and industry representatives will evaluate the relevance of the total curriculum and make recommendations for additions, deletions or revisions at the course level.

B) At the course level, courses and modules funded by RRSCS will be evaluated by a faculty and student peer review team on an annual basis after the first offering and then reviewed every other year. The Ohio Learning Network's rubrics for online courses will be used as part of the evaluation process.

C) Courses taught by RRSCS faculty, but not funded by the School, will be examined as part of the overall curriculum review, and will be evaluated every other year by a faculty and student peer review team. This team shall contain some members from the review teams created in paragraph B above.

STUDENT SERVICES EVALUATION

An annual assessment will be made of student services related to the Ralph Regula School. The School will coordinate this effort with participating campuses to field regular surveys of students and campus administrative staff to ensure that students advising, registration, and other procedures are working satisfactorily from the student's viewpoint.

SIGNATORIES

The undersigned academic officers agree to the terms of this consortial agreement for the Ralph Regula School of Computational Science.

Ronald L. St. Pierre
Acting Provost
Capital University

Joseph A. Alutto
Interim Provost
The Ohio State University

Toy Caldwell-Colbert
Provost
Central State University

Anthony J. Perzigian
Provost
University of Cincinnati

Kay Adkins
Provost
Columbus State Community College

Kenneth W. Bladh
Provost
Wittenberg University

Robert G. Frank
Provost
Kent State University

Steven R. Angle
Provost
Wright State University

Helen Grove
Provost
Sinclair Community College

Ralph Regula School of Computational Science Application for Host Institution Class Enrollment

Complete all items below. Sign and return the form to the registrar's office of your **home** institution. You will be registered for the class(es) at the host institution you have indicated below. You are responsible for observing all host institution registration, drop/add and withdrawal deadlines. **If enrolling in classes at more than one host institution, you must complete and submit a separate form for each.**

Name: _____ **Social Security Number:** ____/____/____
Last First Middle

Other name(s) used previously: _____ **Date of Birth:** ____/____/____
Month Day Year

Permanent Mailing Address: _____
(Number and street. If P.O. Box, number and street also required)

City State/Country Zip

Telephone: _____ **E-mail Address:** _____
(include area code)

Gender: Female Male **Country of Citizenship:** _____

Check if this applies:: I am a Permanent resident alien of the U.S. **or** Refugee **or** Asylee

Alien/File # A _____ Date approved: ____/____/____
Month Day Year

My most recent Visa type is: _____ Date issued: ____/____/____
Month Day Year

Residency: Students enrolled in a public institution will retain their residency status as determined by their home institution. Students enrolled in a private institution will be reviewed for a residency determination.

Most recent high school attended: _____ **Graduation date:** _____
Name of High school City State

Prior College Degree & Institution (if applicable): _____

Current Home Institution: _____
Institution for your primary registration and degree program

Class(es) in which you are enrolling:

1) _____	_____	_____	_____
Home Course Number and Class Section	Host Course Number and Class Section	Host Institution Term & Year	Host Institution Credit hours
2) _____	_____	_____	_____
Home Course Number and Class Section	Host Course Number and Class Section	Host Institution Term & Year	Host Institution Credit hours

I affirm that the information I have provided on this application is complete and accurate. Pursuant to the Family Educational Rights and Privacy Act of 1974, as amended (FERPA), I hereby authorize both my home institution and the host institution from which I am taking this course to exchange registration and required financial aid information regarding my enrollment in the class(es) noted on this form. I also authorize the host institution to release an official copy of my host institution transcript at the end of the term to my home institution. I understand my home institution will add the course(s) to my transcript and charge me the appropriate tuition and fees for the additional credit hours. I also agree to abide by the class schedule and course drop and/or withdrawal dates associated with the host institution from which I am taking this course.

Applicant's Signature _____
Date

To Be Completed by the Home Institution

Home institution residency status: In-state Out-of-state

I hereby certify that student named above is in academic good standing with this institution and is authorized to enroll in the host institution class(es) as indicated. Institution: _____

Certifying Official: _____ **Title:** _____
Please Print

Telephone: _____ **Fax:** _____ **E-mail Address:** _____

Signature: _____ **Date:** _____

Appendix B

OSU Course Offerings and Prerequisites

Course Number and Title	Prerequisites	Quarters offered	Credits	Topic
Math 151: Calculus and Analytic Geometry I	Mathematics 150 (with grade C- or better) or satisfactory score on Ohio State Math Placement Test	Autumn, Winter, Spring, Summer	5	Calculus
Math 152: Calculus and Analytic Geometry II	Mathematics 141 or 151	Autumn, Winter, Spring, Summer	5	
Math 153: Calculus and Analytic Geometry III	Mathematics 152	Autumn, Winter, Spring, Summer	5	
Math 161: Accelerated Calculus with Analytic Geometry I	Math 151 or permission of dept.	Autumn	5	
Math 162: Accelerated Calculus with Analytic Geometry II	Mathematics 161 or written permission of Math Counseling Office	Winter	5	
Math H190: Elementary Analysis I	Permission of dept.	Autumn	5	
Math H191: Elementary Analysis II	Mathematics H190 with a grade of C or better or written permission of Honors Committee chairperson	Winter	5	
ChBE 790 Process Modeling and Simulation	Permission of instructor	Autumn	3	Simulation & Modeling
CEG 640 Civil and Environmental Systems Engineering	Permission of instructor	Autumn	4	
CSE 778 Computer-Aided Design and Analysis of VLSI Circuits	CSE 560; ECE 561; 601; 675 or ECE 662	Autumn	4	
ISE 521 Operations Research I: Simulation of Production Systems	ISE 500; Stat 427 and 428, or equiv	Au, Wi	5	
ISE 704 Introduction to Discrete System Simulation	Stat 426 or 428	Winter	4	
ME 785 Modeling, Simulation and Control of Hybrid-Vehicles	ME 784 or permission of instructor	Winter (Offered even numbered Years)	4	
MSE 533 Modeling of Materials Processing Methods	MSE 525 and 526, Matsc&en major or permission of instructor	Spring	3	

Course Number and Title	Prerequisites	Quarters offered	Credits	Topic
CSE 202 Introduction to Programming and Algorithms for Engineers and Scientists	Math 151	Su, Au, Wi, Sp	4	Programming & Algorithms
CSE 221 Software Development Using Components	Math 151 or 161/H161 or H190; 201 or 202 or 203 or 204 or En Graph 167 or Engineer H192 or CS&E Placement Level A	Su, Au, Wi, Sp	4	
CSE 294P Computational Thinking in Context: Science and Engineering	Math 151 or equivalent	Sp	4	
EG 167 Problem Solving through Programming for Engineering Calculations and Computer Graphics	Math 151 or equivalent	Su, Au, Wi, Sp	4	
AAE 581 Numerical Methods in Aerospace Engineering	En Graph 167, 580	Autumn	3	Numerical Methods
CEG 406 Professional Aspects of Civil and Environmental Engineering	Civil en or Env Eng major; must be taken as soon as possible upon entering the major	Wi, Sp	1	
CSE 541 Elementary Numerical Methods	CSE 221/H221 or 230 or 502; Math 153	Su, Au, Wi, Sp	3	
ECE 715 Introduction to Numerical Methods for Electromagnetics	ECE 301, and Math 568 or 571; or grad standing	Autumn	3	
Math 606 Introduction to Numerical Analysis of Partial Differential Equations	Math 512 and 572 or equiv with permission of instructor	Sp	3	
Math 607 Essentials of Numerical Analysis	Math 548 or 652 or permission of the Graduate Studies Committee	Wi	5	
ME 250 Numerical Methods and Analysis in Mechanical Engineering	Math 415 or 255; and enrollment in engineering major	Au, Wi, Sp	4	Optimization
CBE 781 Chemical and Biomolecular Engineering Optimization	En Graph 167 or equiv or permission of instructor	Sp	3	
CEG 776 Network Algorithms in Transportation Systems	CEG 405, 540, and EG 167 or CSE 201 or 221 or equiv.	Wi	5	
ECE 759 Numerical Optimization for Electrical Engineers	ECE 352	Au	3	
ISE 522 Operations Research II: Fundamentals of Linear Optimization with Applications	ISE 500, Math 254, 415 or 255, and 568 or 571. Working knowledge of Excel	Au, Wi	3	
ME 761 Optimization in Mechanical Design	ME 562 or 563 or permission of instructor	Spring	3	
MSE 600 Materials Selection and Performance I	Sr standing in MatSc&Eng or Ceram Eng or Metal Eng or permission of the instructor	Wi	3	

Course Number and Title	Prerequisites	Quarters offered	Credits	Topic
CE 660 Civil Engineering Capstone Design	Sr standing. Must be taken as close to graduation as possible	Au, Wi, Sp	4	Capstone
CSE 699 Undergraduate Research in Computer Science and Engineering		Su, Au, Wi, Sp	1-5	
CSE H783 Honors Research	Honors standing; permission of instructor	Su, Au, Wi, Sp	1-5	
ME 564 Senior Design Group Project	ME 510; a second writing course and prereq or concur: 563	Au, Wi, Sp	3	
ME 565.01 Mechanical Engineering Design	ME 562 and 2nd writing course, 3.4 or higher GPA or permission of instructor	Au	3	
ME 565.02 Mechanical Engineering Design	ME 565.01	Wi, Sp	3	
ME 565.03 Mechanical Engineering Design	ME 565.02	Au, Sp	3	
MSE 695.01 Senior Design Project I	Sr standing in MSE or the physical sciences	Au, Wi, Sp	1	
MSE 695.02 Senior Design Project II	MSE 695.01	Au, Wi, Sp	1	
MSE 695.03 Senior Design Project III	Sr standing in MatSc&En and MatSc&En 695.02	Au, Wi, Sp	1	
CE/ECE 675 Instrumentation, Signals, and Control in Transportation Applications	Elec Eng 301 and Math 415; or Civil En 570; or grad standing in elec eng or civil eng	Au	3	Discipline Specific Courses
CSE 630 Survey of Artificial Intelligence I: Basic Techniques	CSE 222/H222 or 230 or 502; Math 366	Au, Wi, Sp	3	
CSE 655 Introduction to the Principles of Programming Languages	CSE 560 and 625	Su, Au, Wi, Sp	4	
CSE 660 Introduction to Operating Systems	CSE 560; 675 or ECE 662; Stat 427	Su, Au, Wi, Sp	3	
CSE 670 Introduction to Database Systems I	CSE 314 or 222 or 230 or 502; Math 366	Su, Au, Wi, Sp	3	
CSE 675.01 Introduction to Computer Architecture	360 or ECE 265; Math 366; ECE 261	Au, Wi, Sp	3	
CSE 675.02 Introduction to Computer Architecture	360 or ECE 265; Math 366	Su, Au, Sp	4	
CSE 680 Introduction to Analysis of Algorithms and Data Structures	CSE 560 or 668 or ECE 668; Stat 427; Math 566	Su, Au, Wi, Sp	3	
Chem 644 Computational Chemistry	Chem 252	Au	3	
MSE 756 Computational Materials Modeling	Permission of instructor	Au	3	
Phys 780				

Course Number and Title	Prerequisites	Quarters offered	Credits	Topic
CSE 621 Introduction to High-Performance Computing	CSE 541; Math 568 or Math 571 or Math 601. Course is well suited to grad students from science/engineering in addition to CS&E students	Au	3	Elective
CSE 694L Scientific Vizualization		Su, Au, Wi, Sp	1-5	
Math 255 Differential Equations and Their Applications	Math 254	Su, Au, Wi, Sp	5	
Math 415 Ordinary and Partial Differential Equations	Math 254	Su, Au, Wi, Sp	4	
Math 568 Introductory Linear Algebra	Math 254 or equiv with written permission of dept	Su Term 1, Au, Wi, Sp	3	
Math 571 Linear Algebra for Applications I	Math 254	Su Term 1, Au, Wi, Sp	3	



Department of Materials Science and Engineering

177 Watts Hall
2041 College Rd.
Columbus, OH 43210
phone: 614-292-6085
fax: 614-292-9857
buchheit.8@osu.edu
mse.osu.edu

To: P. Sadayappan
From: Rudy Buchheit
Re: UG Minor in Computational Science
Date: February 17, 2009

The Department of Materials Science and Engineering is in agreement with the following courses being used for the Undergraduate Minor in Computational Science. With the exception of the 695 series, each of the courses is offered yearly, and may be taken with permission of the instructor. The MSE 695 series is limited to MSE majors only.

MSE 533 Modeling of Materials Processing Methods	MSE 525 and 526, MSE major, or permission of instructor	Yearly Spring	3
MSE 600 Materials Selection and Performance I	Rank 4 MSE major or permission of the instructor	Yearly Winter	3
MSE 695.01 Senior Design Project I	MSE Majors only	Offered each quarter	1
MSE 695.02 Senior Design Project II	MSE 695.01	Offered each quarter	1
MSE 695.03 Senior Design Project III	MSE 695.02	Offered each quarter	1
MSE 756 Computational Materials Modeling	Permission of instructor	Yearly Autumn	3



Department of Integrated Systems Engineering

210 Baker Systems Building
1971 Neil Avenue
Columbus, OH 43210-1217

Phone (614) 292-6239
Fax (614) 292-7852

20 April 2009

Professor P. Sadayappan
Department of Computer Science and Engineering
Campus

Re: Minor Program in Computational Science

Dear Professor Sadayappan:

The Undergraduate Studies Committee of the Department of Integrated Systems Engineering has met and considered the proposal for a new minor in computational science. It is our opinion that this minor program will serve a real need among our undergraduate students, and we heartily concur with the proposal. My apologies for the extreme tardiness in responding to your request.

Best regards,

A handwritten signature in cursive script that reads "Blaine Lilly".

Blaine Lilly
Associate Professor
Chair, Undergraduate Studies

Cc: J. Higle, G. Valco, B. Weide



Civil and Environmental Engineering and Geodetic Science

470 Hitchcock Hall
2070 Neil Ave
Columbus, OH 43210-1275

Phone (614) 292-2771
Fax (614) 292-3780

20 April 2009

Professor P. (Saday) Sadayappan
Department of Computer Science and Engineering The Ohio State >
2015 Neil Avenue
The Ohio State University
CAMPUS

Dear Professor Sadayappan:

This letter is in support of the proposed minor program in computational science. We offer the following comments to consider:

- On page 3 of the proposal, under item #2 Programming and Algorithms, there is a statement: "In addition to several existing courses at OSU that would satisfy this requirement..." However, Table 2 only lists three CSE courses. We believe that EG 167 would in theory also apply as a Programming and Algorithms course.
- We recommend that CE/ECE675 – Instrumentation, Signals and Control in Transportation Applications – would be a good candidate for a domain-specific computationally-oriented course for item #6 on page 4 (see also Table 2, p. 7).

The civil engineering courses listed in the proposal – CE406, CE640, CE776, CE/ECE675 (proposed) – can be included in the minor program. CE406 is offered every year. We would likely offer the 600-level and 700-level classes at least every other year, depending on enrollment figures.

In summary, we support the proposed minor program in computational science.

Sincerely,

A handwritten signature in cursive script, appearing to read "Carolyn J. Merry".

Carolyn J. Merry
Professor and Chair

CF: P. Fox, B. Coifman, G. Bohrer



Department of Mechanical Engineering

College of Engineering
Peter L. and Clara M. Scott Laboratory
201 West 19th Avenue Suite N350
Columbus, OH 43210-1142

Phone (614) 292-2289
Fax (614) 292-3163
Email Meadmin@mecheng.osu.edu
www.mecheng.ohio-state.edu

Professor P. Sadayappan
Department of Computer Science and Engineering
The Ohio State University
2015, Neil Avenue
Campus

Dear Saday,

Please let this letter serve as a statement that the Department of Mechanical Engineering agrees to have the following courses listed as options in the Computational Science Minor which your department is proposing. You also asked for the frequency of offering of these courses and what the "hard" prerequisites are. That information is also given below.

<u>Course</u>	<u>Quarters of Offering</u>	<u>Hard Prerequisites</u>
ME 250	Au, Wi, Sp	Math 415 or 255
ME 564	Au, Wi, Sp	ME 510 and 562 ⁽¹⁾
ME 565.01	Au	(2)
ME 565.02	Wi, Sp	ME 565.01
ME 565.03	Au, Sp	ME 565.02
ME 761	Sp	ME 562
ME 785	Wi (every other year)	ME 784 ⁽³⁾

NOTES:

(1) While this is a capstone course and the emphasis is on design, there should be either some thermal or stress analysis involved, depending on the kind of project. Students should have had some experience with both kinds of analysis to approximately the level of ME 510 and ME 562.

(2) First, all sections of this course except one focusing on biomechanical design projects have been discontinued. This course concept has instead been shifted to the College of Engineering. The new courses are ENG 658 and ENG 659.01 and ENG 659.02. Bob Rhoads of the new Engineering Education Innovation Center is the contact for these courses. You should contact him about these. Second, as far as prerequisites go the first course ME 565.01 probably requires just the maturity of senior standing in engineering or a technical field, but just as in ME 564 the actual project part of the course ME 565.02 and ME 565.03 should require thermal or stress analysis of parts.

Professor P. Sadayappan, page 2

(3) ME 785 is directly tied to ME 784. ME 784 presents the energy analysis of hybrid vehicles and ME 785 deals with optimization of these systems using the models developed in ME 784. Experience has shown that without having ME 784 students struggle with ME 785. Undergraduate students in both ME and ECE who have taken both as technical electives have been successful. Hence you should add ME 784 to the list and we would be happy to have more students in these courses. However, all students will have to get in to ME 784 by the Permission of Instructor route. That way we can make sure students will be able to handle the analysis techniques in the course. ME 784 is offered in Autumn quarter before each offering of ME 785 in Winter quarter. The next offerings are Au 09/Wi 10 for the two courses.

Please let us know if you need any further information.

Sincerely,



Dan Mendelsohn, Chair
Undergraduate Studies Committee
Department of Mechanical Engineering
292-2413
mendelsohn.1@osu.edu

cc.

K. Srinivasan
G. Kinzel



Department of Electrical & Computer Engineering

205 Drees Labs
2015 Neil Avenue
Columbus, OH 43210-1272

Phone (614) 292-2572
Fax (614) 292-7596

May 1, 2009

Professor Sadayappan
CSE Department

Dear Professor Sadayappan,

The ECE Curriculum committee, with input from the ECE Computer area committee, has examined the plan for a Minor Program in Computational Science and concurs with the proposal.

We have three comments. One is that as according to the plan itself, when the minor is reexamined periodically, we may have new courses to recommend including in the list of courses. Secondly, for the two ECE courses currently in the list of courses, while we expect those courses to keep running, if enrollment is too low, we reserve the right to cancel their offerings. Lastly, those two ECE courses in the list are not currently organized as distance learning courses. We may be able to convert them to a distance learning format but we can't promise that.

Sincerely,

A handwritten signature in black ink, appearing to read 'CA Klein', written in a cursive style.

Charles A. Klein
Professor and Associate Chair for Instruction



Department of Computer Science and Engineering

395 Dreese Labs
2015 Neil Avenue
Columbus, OH 43210-1277

Phone (614) 292-5813
Fax (614) 292-2911
www.cse.ohio-state.edu

To: P. Sadayappan

From: Bruce W. Weide, Professor and Associate Chair, CSE Department

A handwritten signature in black ink, appearing to read "Bruce W. Weide".

Date: 21 April 2009

Re: Computational Science Minor Proposal

I am happy to provide CSE Department endorsement of your proposal, and to answer the specific questions about course offerings on the attached page. Please let me know if you have any questions.

Course	Offerings	"Hard" Prerequisites
CSE 202: Introduction to Programming and Algorithms for Engineers and Scientists	Au, Wi, Sp, Su	Math 151 (intro calculus)
CSE 221: Software Development Using Components	Au, Wi, Sp, Su	One prior programming course or equivalent experience
CSE 294P: Computational Thinking in Context: Science and Engineering (intended permanent course CSE 205)	Au, Wi, Sp	Math 151 (intro calculus)
CSE 541: Elementary Numerical Methods	Au, Wi, Sp, Su	Two prior programming courses or equivalent experience; Math 153 (multi-dimensional calculus)
CSE 621: Introduction to High-Performance Computing	Au	Two prior programming courses or equivalent experience; CSE 541 or ME 250 or equivalent (numerical methods); Math 658 or equivalent (linear algebra)
CSE 630: Survey of Artificial Intelligence I: Basic Techniques	Au, Wi, Sp	Two prior programming courses or equivalent experience; Math 366 or equivalent (intro discrete math)
CSE 655: Introduction to the Principles of Programming Languages	Au, Wi, Sp, Su	CSE 560 or equivalent (large programming project); CSE 625 or equivalent (intro theory of computation)
CSE 660: Introduction to Operating Systems	Au, Wi, Sp, Su	CSE 560 or equivalent (large programming project); CSE 675 or ECE 662 or equivalent (intro computer architecture)
CSE 670: Introduction to Database Systems I	Au, Wi, Sp, Su	Two prior programming courses or equivalent experience; Math 366 or equivalent (intro discrete math)
CSE 675: Introduction to Computer Architecture	Au, Wi, Sp, Su	Two prior programming courses or equivalent experience; CSE 360 or ECE 265 or equivalent (intro computer organization)
CSE 680: Introduction to Analysis of Algorithms and Data Structures	Au, Wi, Sp, Su	Two prior programming courses; CSE 560 or equivalent (large programming project); Stat 427 or equivalent (intro probability and statistics); Math 366 or equivalent (intro discrete math)
CSE 694L: Introduction to Visualization	Sp	CSE 541 or ME 250 or equivalent (numerical methods); CSE 581 or equivalent (intro computer graphics)
CSE 778: Computer-Aided Design and Analysis of VLSI Circuits	Au	ECE 561 or equivalent (intro digital circuit design); CSE 675 or ECE 662 or equivalent (intro computer architecture)