CS4360 Spring 2020

Technical Software Project

Program Goals

The MSU Denver Computer Science program enables its graduates to become:

- Professionals capable of applying computer science and software engineering principles and practices
- Graduate students in computer science and related fields
- Life-long learners capable of self study, continuing education and ongoing professional development
- Ethical practitioners in computer science, software engineering and related fields
- Innovators able to respond to technological change and intellectual challenge

Program Outcomes

The program enables students to achieve, by the time of graduation:

- An ability to apply knowledge of computing and mathematics appropriate to the discipline;
- An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution;
- An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs;
- An ability to function effectively on teams to accomplish a common goal;
- An understanding of professional, ethical, legal, security, and social issues and responsibilities;
- An ability to communicate effectively with a range of audiences;
- An ability to analyze the local and global impact of computing on individuals, organizations and society;
- Recognition of the need for, and an ability to engage in, continuing professional development;
- An ability to use current techniques, skills, and tools necessary for computing practices.
- An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
- An ability to apply design and development principles in the construction of software systems of varying complexity.

Course Description

This course provides an experience in working on a software development project that requires technical knowledge. Students will work in teams of 3 or 4 to identify a problem, design a solution to that problem, and implement that solution. The solution must involve creating software and may involve also creating hardware.

Student Learning Objectives

1. Identify a real-world problem amenable to solution by software, perhaps with accompanying hardware.
2. Design a software solution to a given problem.
3. Analyze the feasibility of implementing a given software design.
4. Apply technical knowledge gained from courses and readings in the technical literature to a project.
5. Collaborate effectively as a team member.
6. Analyze and reflect, verbally and in writing, on a software development process.

Outline

1. Identifying Real-World Problems
    1. Analyzing Legal, Social, and Ethical Impacts of a Proposed Solution
    2. Estimating the Feasibility of Solving a Problem
2. Designing a Software Solution to a Problem
3. Researching Technical Resources to Develop a Solution
4. Implementing a Proposed Solution in Software
5. Refining the Scope of the Problem
6. Testing and Debugging the Software
7. Presenting the Project

Communication

Face to face communication is best. I don't answer my phone. We will use the #cs4360 channel at https://drb80.slack.com/signup for asynchronous communication about the class. If you need to use the MSU Denver official communication channel: beatys@msudenver.edu

Prerequisites

Senior standing and CS 3210, CS 3600, 8 additional credits of upper-division CS courses, COM 2610, PHI 3370, and CAS 1010, all with grades of "C-" or better, or permission of instructor.

Schedule

Time	Monday	Tuesday	Wednesday	Thursday
1100 -- 1200	Office Hour AES 200N		Office Hour AES 200N	
1200 -- 1400	CS 4360 AES 210		CS 4360 AES 210	
1500 -- 1600	Office Hour AES 200N		Office Hour AES 200N	
1600 -- 1800	CS 3250 AES 210		CS 3250 AES 210	
1730 -- 1830		Office Hour AES 200N		Office Hour AES 200N

Time	Monday	Tuesday	Wednesday	Thursday
1830 -- 1945		CSS 2751 AES 490		CSS 2751 AES 490

Dates

Make sure you check the official university calendar for accuracy.

Day	Date	Event
Tue.	Jan 21	Classes begin
Mon.	Jan 27	Self add deadline
Tue.	Jan 28	Last day to drop full-term courses with 100% refund
Thu.	Feb 6	Last day to drop full-term courses with 50% refund and have classes deleted from student record.
	Mar 23 - 28	Spring Break
Wed.	Sept 4	Last day to register for full-term classes with department permission
Fri.	Apr 3	Last day to withdraw from full-semester classes and receive a 'W'
Sat.	May 9	Last day of classes
	May 11 - 16	Finals Week

Credit Hours

Federal Credit Hour Definition: A credit hour is an amount of work represented in intended learning outcomes and verified by evidence of student achievement that is an institutionally-established equivalency that reasonably approximates not less than one hour of classroom or direct faculty instruction and a minimum of two hours of out-of-class student work each week for approximately fifteen weeks for one semester. This equates to 45 hours per credit per semester, or 180 hours for this class.

Required Books

Clean Code: A Handbook of Agile Software Craftsmanship 1st Edition
Robert C. Martin
ISBN-13: 978-0132350884
ISBN-10: 0132350882

The Pragmatic Programmer: From Journeyman to Master
by Andrew Hunt and David Thomas

Getting Things Done: The Art of Stress-Free Productivity
March 17, 2015
David Allen

Writing That Works; How to Communicate Effectively In Business
Kenneth Roman, Joel Raphaelson
ISBN-13: 978-0060956431
ISBN-10: 0060956437

Grading

This is a computer science senior experience course. All senior experience courses must demonstrate students abilities to do the following.

1. Synthesize learning through critical analysis and logical thinking.
    - Students must invent or recognize an appropriate problem for their project for this course, including critical analysis of social and ethical implications of their proposed artifact.
    - The project must be challenging enough, as judged by the instructor, that they will need to synthesize learning from a number of previous courses, including courses in computer science, mathematics, and science, in order to successfully complete it.
    - The requirement that they reflect on their on-going process and present their final product will require them to use expertise gained from the technical writing course and the public speaking course.
2. Apply theoretical constructs to practical applications.
   The core requirement of this course is that students use technical knowledge, much of which is theoretical in nature, to produce a software artifact that will solve a practical, real-world problem.
3. Critique philosophical tenets and current practices.
   In analyzing possible problems, and later in examining possible solutions to arrive at the approach they will implement, students will need to consider the social and ethical impact of solving their proposed problem, and they will need to critique existing solutions to the problem.
4. Integrate and refine oral and/or written communication skills.
   Throughout the course, students will be strengthening their oral and written communication skills by working in a team, writing and presenting reflections on their on-going development processes, and presenting their final project.
5. Verify their expertise.
   The instructor will ensure that the problem proposed by the students presents an appropriate technical challenge, and the students will be required to produce a functioning solution to their problem, or an appropriately narrowed version of their problem, by the end of the course.

To pass this class, you must demonstrate you have each of the five skills listed above.

For this four-credit class, 180 hours of work is required. You should expect to work 10 hours outside of class each week of the semester.

To pass this class, you must demonstrate that you have performed 180 hours of work.

To pass this class, you must take the CS Major Field Test.

To pass this class, you must fill out the senior survey.

To pass this class, you must make valuable contributions to your team's success. This

typically is measured by stories completed and commits to your team's repository.

Methods of evaluation.

1. Peer and instructor evaluation of performance as a team member on the project
2. Customer evaluation of the functionality of the project
3. Oral and written reflections on the development process
4. Oral and written presentation of the completed project

There will be two tests that count for 10% each.

Letter Grade	From %	To %
A	90	100
B	80	89
C	70	79
D	60	69
F	0	59

Cheating

You must not claim the work of others as your own. If you use the work of others you must properly cite it. In programs, modifications to others programs such as renaming variables, reordering methods, etc., is still cheating. You must state who helps you on a program, and they must state that they have helped you. If you cheat once, you will receive a zero on that assignment or test; if you cheat twice you will be reported to the department chair and Student Conduct in the Dean of Students Office and will result in at least an F in this class. If you have any questions as to what constitutes cheating, please contact me. Please see https://www.plagiarism.org/

Getting Help

In general, I recommend against Googling for answers to problems you have in the class. There is an inordinate amount of bad information about computing on the Internet. I recommend using the discussion board for this class, learning assistants, and me for the questions you have.

Submitting Assignments

You will submit your assignments online at the class's Moodle site. If you haven't used moodle before, you must self enroll at https://gouda.msudenver.edu/moodle/ This class's site is: https://gouda.msudenver.edu/moodle/course/view.php?id=211 The enrollment key for this class is "forearm deserve victoria visiting". Assignments will typically be due Sunday night at midnight. There is a 10% per day late penalty. My help before the assignment is due cost nothing; after it is due it is 5 points a minute. You must submit all assignments to pass this class.

Classroom Behavior

This is a university classroom: treat it as such. Any disruption will cost you 10 points the first

time, 50 points the second, and will earn you an 'F' the third time. If you have a question, ask it in class as it is my experience that if you have a question, most of the other students do too. If you need to talk with a classmate, take it outside or online.

Attendance Policy

Do not attend class unless you are ready and willing to learn. You are responsible for everything presented or discussed in class.

University Policies

Students are responsible for full knowledge of the provisions and regulations pertaining to all aspects of their attendance at MSU Denver, and should familiarize themselves with the policies found in the MSU Denver Catalog.

For more information and recent updates, go to the CLAS website