**CISC 480 - Syllabus\***

Section 01, Spring 2019

*University of St. Thomas*

*Department of Computer and Information Science)*

*\*(This is a living document, changes will not be made without informing the class)*

**Instructor**: Joseph Myre, Ph.D.

**Office**: OSS 410

**Email**: [myre@](mailto:joemyre@uark.edu)stthomas.edu

**Office Hours**: M 1045 – 1145

T 1035 – 1135

Or by appointment

**Course Number:** CISC 340, Section 01

**Term:** Spring 2019

**Course Title:**

Senior Capstone

**Course Meeting Schedule**:

Lecture: MWF 1335 – 1440, OSS 432

**Catalog Description**:

The senior capstone course provides computer science majors the opportunity to integrate the knowledge that they have gained from across the curriculum. Students will work in groups to design, document and implement a large sized software project. During this process, students will be exposed to programming team organization, software development practices, as well as tools that facilitate the development of software systems. Additionally, students will explore the historical development of computer science through readings and discussions.

**Prerequisites:**

1. Senior Standing, 2) C- in CISC 340, 350, and 380.

**Final Exam:**

Wednesday, May 22, 1030 - 1230

**Canvas:**

I will attempt to use canvas (canvas.stthomas.edu) to post announcements, lecture notes, assignments, and other useful course documents. It is your responsibility to frequently check Canvas.

**Textbook:**

There is no assigned text for this course. Articles and other handouts will be provided in class.

**IDEA Learning Objectives for this course:**

* Understand the role of key elements of software engineering such as requirements

analysis, software design, documentation and software testing

* Understand the principles of modern software processes
* Understand and navigate the complexities of managing and working on a team to

complete a medium to large software project

* Be able to apply the principles of object-oriented analysis and design, and the

corresponding UML notation

* Be able to apply principles and patterns for software design
* Be familiar with the characteristics of a good API
* Be familiar with the standard techniques for software testing
* Be able to research and compare tools that facilitate the development of software
* Be exposed to some of the historical development of computer science through

readings and discussions

**Methods for Assessing Learning Objectives:**

The expected learning outcomes for the course will be assessed through one group project, homework assignments, in-class activities, and participation in class discussions.

**Group Project:**

A key component of this course is a group project. The project is designed to give students the chance interact with the material present in their major courses and apply it to a large sized software project. Developing particular piece of software is not the primary goal of this project, but rather it is a vehicle for students to demonstrate, develop and hone their software engineering skills.

The project also introduces the idea of developing software in teams. Most modern software companies employ a team structure to develop their products. Class teams will consist for 3-5 students. All team members must take part in all project activities and will be responsible for knowing how all parts of the project. However, responsibilities may be

divided so different members take the lead in different activities. Though the scope of

individual effort on a single component of the project may vary, in the end all group

members should have contributed equally to the project.

The projects will consist of 5 deliverables: initial plan, an intermediate report, an

intermediate demo, a final report, and a final demo. The specific requirements of each

deliverable will be discussed in class. However, professional presentation is a requirement for all parts of the project. Students are encourage to create their proposals and reports using Latex and incorporate slide presentations to accompany their demos.

**Attendance and Classroom Conduct:**

Attendance at all lectures and labs is expected and highly recommended. I will give significant hints and help during lab session. I expect that any student who comes to lecture or lab is there to learn and will not talk on the phone, text, or browse the Internet during class. These activities likely bother the students sitting nearby, so any student caught doing them will be asked to leave.

If at all possible, please let your instructor know in advance if you are going to be absent. Tests may be made-up for excused absences outlined in the University Student Manual. See the official student rules for clarification of excused absences. Students are responsible for obtaining missed lecture notes and other information, such as announcements and assignment due dates, from their colleagues in the course.

**Workload Expectations:**

As this is a 4 credit course, in addition to the three 65-minute class periods, the average student will be expected to spend approximately 2 or more hours outside of class for each hour in class. Outside-of-class activities include finishing assignments, reading the textbooks, reviewing and tweaking programs we write during lecture, getting help at office hours, working on the practice quizzes, and other general studying and programming practice.

**Collaboration Policy and Academic Dishonesty**:

Unless otherwise noted, you must work individually on each assignment. For assignments requiring pairs, you and your partner should only submit one set of solutions for the group that clearly indicates who the group members are. Both group members must collaborate through the entire process; groups are forbidden from “splitting up” the problems. Both group members will receive an identical score.

**Important: You are not allowed to discuss problems with anyone other than your other group member or the instructor. If you work individually on an assignment, you may only discuss problems with the instructor. Any violation of this policy is considered academic dishonesty.**

Most of the assignments involve writing computer code. It is unacceptable to simply copy another student’s/group’s solution or code for a problem and claim it as your own; this will be treated as academic dishonesty for all parties involved, including the student/s who provided the code. Copying code from the Internet is also considered academic dishonesty.

As a core part of its mi**s**sion, the University of St. Thomas provides students with the opportunity to further their educational goals through programs of study and research in an environment that promotes freedom of inquiry and academic responsibility. Accomplishing this mission is only possible when intellectual honesty and individual integrity prevail. Each University of St. Thomas student is required to be familiar with and abide by the University’s ‘Academic Integrity Policy’ which may be found at: https://www.stthomas.edu/policies/undergraduate/disciplinaryrights/academic/. If you have not read these policies you are encouraged to do so.

A student caught committing an act of academic dishonesty will be reported to the Dean of the College of Arts and Sciences. Cheating is taken very seriously at St. Thomas. Any academic dishonesty will be reported to the Academic Integrity Monitor, and appropriate sanctions will be determined by the University. At minimum you will receive a score of zero on the assignment in question and a 20% deduction on their overall course grade. Further penalties are at the discretion of the instructor. Other penalties could involve failing the class or suspension from the university.

**Calculation of Grades**

5%: Meeting Notes

5%: Initial plan

10%: Intermediate project report

10%: Intermediate project demo

15%: Final project report

15%: Final project demo

10%: Manager reviews

30%: Homeworks

**Late Assignments**:

Late assignments will not be accepted.

**Re-grades**:

If you think an error was made in the way your assignment or exam was graded, you have one week from the day the graded lab or exam was returned to you to submit a request for a re-grade. This request should be an email sent to Professor Myre clearly outlining why you think you should receive more points. For exam re-grades, you should also return your graded exam to Professor Myre within one week of receiving the graded copy. Any re-grade requests received more than one week after the graded item is returned will not be considered.

**Grading Scale:**

There is no curve; final grades are assigned as follows.

|  |  |
| --- | --- |
| **Final Percentage** | **Letter Grade** |
| 93-100% | A |
| 90-93% | A- |
| 87-90% | B+ |
| 83-87% | B |
| 80-83% | B- |
| 77-80% | C+ |
| 73-77% | C |
| 70-73% | C- |
| 60-70% | D |
| Below 60% | F |

**Important:** Scores will be assigned exactly corresponding to the above table without any exceptions. It is possible that the boundaries between letter grades will be adjusted ONLY in a manner that is advantages for the students. Do not assume this will happen, it is only a possibility. Scores will NOT be rounded up under any circumstances. Please do not email the instructor at the end of the semester asking if he will bump you up to the next grade level; he will not. The instructor will give ample help during office hours, so there are plenty of opportunities throughout the semester to make sure you get the grade that you desire.

**Americans with Disabilities Policy**:

The American Disabilities Act (ADA) is a federal antidiscrimination statute that provides comprehensive civil rights protection for persons with disabilities to be guaranteed a learning environment that provides for reasonable accommodation.

Students are invited to contact the Disability Resources office about accommodations early in the semester. Telephone appointments are available to students as needed. Appointments can be made by calling 651-962-6315 or 800-328-6819, extension 6315. You may also make an appointment in person in Murray Herrick, room 110. For further information, you can locate the Disability Resources office on the web at http://www.stthomas.edu/enhancementprog/.

**Academic Honesty**:

As a core part of its mi**s**sion, the University of St. Thomas provides students with the opportunity to further their educational goals through programs of study and research in an environment that promotes freedom of inquiry and academic responsibility. Accomplishing this mission is only possible when intellectual honesty and individual integrity prevail. Each University of St. Thomas student is required to be familiar with and abide by the University’s ‘Academic Integrity Policy’ which may be found at: https://www.stthomas.edu/policies/undergraduate/disciplinaryrights/academic/.

Any academic dishonesty will be reported to the Academic Integrity Monitor, and appropriate sanctions will be determined by the University.

**Inclement Weather Policy**:

In the event of severe weather the University may announce that the inclement weather policy is in effect. You may call 651-962-7669 for announcements. In the case of University closure, class will be canceled. If class is otherwise canceled students will be notified by email from the instructor.

**Additional Course Policies:**

Students are responsible for knowing all materials presented in class including lectures, handouts, assignments, reading assignments, etc.

Lectures will begin and end on time. Please do your best to get to class before the start of lecture.

Attendance is strongly encouraged. The instructor is not responsible for material that was missed due to absence. You will have to obtain notes from your peers.

The use of cell phones and laptops will not be permitted in class unless specific permission is given. If you are observed using these devices in class without permission, you will earn demerit points.

Audio or visual recording of class lectures and/or activities is not allowed unless express permission is given by the instructor.

Exams will be given in class and are closed book, closed note.

No makeup exams will be given without extremely extenuating circumstances.

**Topics\*:**

* Overview of software engineering
* Software processes: waterfall model, XP, agile, etc.
* Requirements analysis: use cases, functional, non-functional, sequence diagrams
* Domain modeling: conceptual classes, attributes, associations
* UML, Object-oriented design
* Design patterns
* Software testing
* Development tools

\* - This list is tentative and may expand or contract according to the needs of the class and the pace at which we are able to cover material.