Computer Science 222, Software Engineering

Student Syllabus **Fall Semester 2020**

Professor Gary Locklair, Ph.D.
Office: S112D [CUW] voice mail: (262) 243-4217
email: locklair@cuw.edu gary.locklair@cui.edu
Hours: online by appointment

CSC 222, Section 1 meets online via zoom MW 5:00-6:15pm
https://cui-edu.zoom.us/j/92651592442

CSC 222 - Software Engineering is the management of the entire software development process. This course affords the student the opportunity to explore the art and science of professional software development in great detail. The foundational aspects of the creative process, idea, implementation, and interaction are investigated in the context of software development. Principles of requirements, specifications, design, implementation, and maintenance are studied. The software development lifecycle is used as a management tool for the professional creation of effective systems. Support and management issues including design patterns, user and developer documentation, coding tools, and quality assurance are investigated. Actual programming projects are analyzed along with current research in the field. Several major software projects, both individual and team, are synthesized by students using an industry methodology.

3 credit hours

Text

Software Engineering - The Idea, Implementation, and Interaction of Professional Software Creation © Dr Gary Locklair is available without cost online

Blackboard will contain all course material (including text readings)

Objectives

At the conclusion of this course, you will:

° appreciate the *grand ideas* of computer science by understanding fundamental software engineering concepts.

° be able to carry out software development projects in a logical, consistent, and self-managed manner in accordance with accepted methodology following a software development lifecycle model.

° become a proficient and disciplined software developer, able to construct and meet development milestones.

° appreciate the benefits of proper analysis and design prior to the commencement of the implementation phase.

° develop skills in creating useful user and developer documentation and appreciate their importance.

° develop analysis skills as actual programming processes are studied and lessons are elucidated.

° understand the historical development of software engineering and be familiar with the important research contributions in the field.

° gain insights into the unique properties of both individual and team effort software development.

° enhance programming skills though actual program synthesis.

° comprehend the vital role of software quality assurance, including documentation, testing, verification, and validation.

° enhance technical communication skills by doing both oral and written reports for project related milestones.

° appreciate the effect of a Christian worldview on the discipline of computer science and gain an understanding of the responsibilities and ethical issues a Christian professional will face in a computerized society.

Course Overview

There are three important components to CSC 222: Theory, Practice and Application. **Theory** deals with computer science concepts. For example, the various techniques for code testing (black box, white box, developer versus QA, etc.) will be studied with an emphasis on: what it is, how it works, and why it is used. **Practice** deals with use, that is, exposure to actual systems. For example, production systems (programs) will be analyzed and critiqued with an eye on "potential lessons." **Application** deals with putting concepts to use in order to solve problems; that is, software construction. For example, a large-scale software project will be designed and implemented using a professional software engineering process.

It is expected that you will read and understand the text as the foundation for class discussion. Appropriate text chapters should be read *before* class discussion. Class discussion is based upon the text material but always goes far beyond what is covered in the text. *You* are responsible for understanding the material and following course instructions.

Course Requirements

I will determine how well you have met the objectives by (tentatively):

° unit examinations and quizzes (theory) - 50 percent of final grade
° Homework (theory, application) - 20 percent of final grade
° System Projects (practice, application) - 30 percent of final grade

Discussion

During class time, students are expected to be active participants in discussion. Your responses and questions are necessary and important. Without student feedback, I cannot judge the effectiveness of the current instructional approach. Student responses to my questions bring a vital ingredient to class, namely a different perspective. Your classmates may understand your line of reasoning better than mine! Please don't view questions as unimportant "parroting" of textbook answers; instead use discussion as a true learning experience. Note: lack of participation (eg, inattentive behavior) will result in a lower course grade. Quality participation may increase your course grade.

Exams

Three unit examinations will be given. Each unit exam will contain 3 sections: True/False (20 questions @ 1 point), Multiple Choice (15 questions @ 2 points), and Short Answer (4 to 6 questions @ 4-7 points). Exams are closed book, notes, and technology. Each exam will cover material from the preceding unit. Exam dates and contents are listed on the tentative schedule. If you will miss an exam period, you must leave a message *beforehand*; arrangements may include dropping the exam or taking a makeup exam. (Makeup exams must be taken before the next scheduled class period.) No early exams will be given. Failure to inform me beforehand or to make arrangements immediately after a missed exam may result in an exam score of 0. Cheating on an exam will result in a course failure.

Exam material comes primarily from class discussion. Topics developed during class time will constitute the majority of exam questions. In addition, concepts and skills developed during laboratory sessions will be covered in the exams. Text material will also be included on exams. Finally, some exam questions require critical thinking skills by asking you to synthesize an answer based upon several concepts.

Homework

Homework is assigned at various times throughout the semester. You may use any resource you desire to aid in answering the questions. You may discuss the questions with your classmates; however, plagiarism will not be tolerated. The answers to the questions should be primarily your own work - not a fellow student's. If plagiarism is detected, all parties involved may receive a 0 for that assignment (see the section on *academic ethics* in the student handbook).

I won't provide procedural help for assignments, but I will answer questions about specifications and semantics.

You are responsible for completing and submitting all homework assignments. However, it is possible that not all assignments will be graded.

Late assignments are not accepted.

Logistics

Homework and project assignments are due at the beginning of the period on the due date. Late assignments will not be accepted. (See me regarding emergency situations.)

I will not answer content questions on the day of an exam.

I will place additional information regarding exams, homework, projects, etc. on the *Blackboard LMS*. Check online regularly.

Attendance

Attendance is expected at every class meeting. If you are not present in class, an absence will be recorded on the official attendance form. Class absences will be reported to your advisor. Excessive absences can result in a failure for the class; missing 6 or more class periods will result in a course failure unless prior arrangements have been made. Each class period costs you a bundle in tuition! I want you to get the most for your money. Be here, aloha. Note: students in *diapause* may, at my discretion, be marked as an unexcused absence.

Grading

There are (tentatively) 500 points possible in CSC 222, distributed as:

° 3 unit examinations	- 225 points (75 points each)
° Written Homework	- 100 points total
° Systems Projects	- 175 points total

The following will serve as a tentative grading guide:

90 - 100 percent = A range
80 - 89 percent = B range
70 - 79 percent = C range
60 - 69 percent = D range
less than 60 percent = F (let's try it again next year)

CUI grades with + and - letter grades. Scores in the top two percentage points of any range will receive a plus while scores in the bottom three points of any range will receive a minus.

Computer Lab Time

It is *expected* and *required* that all CSC 222 students will spend an adequate amount of time experimenting with the computing resources and developing solutions or carrying out assignments on their own machines or in the CUI computer labs. The exact amount of time spent in the lab will vary based upon student needs.

Computer Labs

As a 200-level computer science student, you should be familiar with CUI's computing resources already. If not, ask for help. Of course, you can use your own machine instead.

Resources

CUI's Blackboard LMS contains all resources for CSC 222, discussion outlines for each class period, homework assignments, project specifications, exam study guides and many other resources.

Comments From Last Year

Last year, I asked several of the A and B students in CSC 370 at CUW to pass on "words of wisdom" to those who would follow in their footsteps. Here are their comments:

"Try to see the 'big picture.' Memorize the outline and then it's easy to fill in the details. Learn the concepts. Don't dwell in the details."

"Stay focused during discussion; come on, if you don't understand something, ask! Don't get behind on the systems projects. Stay up-to-date with the work."

"Get the outlines from the web site ... don't just follow along during class, but be an active participant. Add your own ideas to the class notes."

"Attend every class. Laugh at his jokes even when it seems impossible. Ask questions. Do the reading ahead of time."

Course Synopsis

Computer science is a scientific and technological subject. In order to understand computer science concepts, you must ... understand other computer science concepts! How do we get started? We will use a 'spiral' approach and look at the subject first from a high level view and then at progressively deeper levels.

Unit 1 – The Idea of Software Engineering

Unit 2 – The Implementation of Software Engineering

Unit 3 – The Interaction of Software Engineering

Some topics:
What is Software Engineering? Why do I care about it? How does it affect my projects? How can projects be improved? How can the process become consistent? How do I create software in a professional manner? Where do we go from here? What can make our jobs easier?

The Framework

For this reason I bow my knees before the Father, from whom every family in heaven and on earth is named, that according to the riches of His glory He may grant you to be strengthened with might through His Spirit in the inner man, and that Christ may dwell in your hearts through faith; that you, being rooted and grounded in love, may have power to **comprehend** with all the saints what is the breadth and length and height and depth, and to know the love of Christ which surpasses knowledge, that you may be filled with all the fullness of God. Ephesians 3: 14-19

Computers can count, but they cannot comprehend. Computer Science is fantastic and fascinating; I enjoy it immensely and I hope that you will too. Computer Science is a creative activity reflecting the nature of the Triune God as we create via *Idea*, *Energy* and *Power*. However, as brightly as computers have changed all of our lives in the past 75 years, they are a black hole compared to the brilliance of the most life changing phenomenon, a personal faith in the Son of God as Savior and Lord.

Very Tentative Schedule

Dates	*Topic and Themes*
8/31 - 9/30	**Unit 1** – The Idea of Software Engineering Definition and description of software engineering Creating tools Software Development Lifecycle Problems with, and improving, software development Teams and tools Mythical Man-Month Requirements and Specifications
9/30	*Unit 1 Exam* [Day 10]
10/5 – 11/9	**Unit 2** – The Implementation of Software Engineering Design and Implementation Design patterns Estimation Testing User Interface No Silver Bullet 12 – 13 October – Reading Days
11/11	*Unit 2 Exam* [Day 20]
11/16 – 12/9	**Unit 3** – The Interaction of Software Engineering Human-computer interaction Computer Scientist as Toolsmith Maintenance Documentation Quality Assurance People and Process 26 – 27 November – Thanksgiving
12/14	*Unit 3 Exam* [Final Exam period]