Overview

This course covers the process of constructing software, including the structural views of software components, and their characteristics and interrelationships, at a high level of abstraction. The course also covers the design principles that govern the purpose, structure, development, and evolution of software components. The informal laboratory component of the course uses software design tools to reinforce design processes and associated design representations.

Prerequisites: CSCI 230. Prerequisite or co-requisite: COMM 104.

Outcomes

After completing CSCI 360 students will be able to:
1. Judge legal issues in software projects
2. Construct UML diagrams
3. Apply UML diagrams in different phases of a software engineering project
4. Formulate the requirements and specifications for a software project
5. Examine software architectural styles
6. Apply appropriate design principles to software projects
7. Build software with OO Design principles
8. Compare different software designs
9. Design and implement software test cases
10. Explain the software lifecycle and how it applies to different projects
11. Build GUIs for enhanced user experience
12. Understand the importance of project management in software
13. Distinguish software design patterns

Materials

Required book:

Reading assignments and exercises will be taken from the textbook.

Other books you may find useful:
“The Mythical Man Month”, Frederick P. Brooks, Jr.. Addison-Wesley, 1972


Software:
1. UML: Visual paradigm
2. Java: Eclipse or NetBeans IDE

Class Meeting times: M/W/F 8:30 – 9:20
Location: HWEA 301

Course Website:
http://mountrouidoux.people.cofc.edu/CSCI360/index.xhtml

We will use Oaks for assignment submission and grading.
**Evaluation**

<table>
<thead>
<tr>
<th>Component</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comprehensive Exam</td>
<td>30%</td>
</tr>
<tr>
<td>Project</td>
<td>30%</td>
</tr>
<tr>
<td>Quizzes</td>
<td>30%</td>
</tr>
<tr>
<td>Homework</td>
<td>10%</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
</tr>
</tbody>
</table>

Your weighted average will result in a letter grade assigned according to the usual scale: A: 93%–100%  
A–: 90%–92%  B+: 87%–89%  B: 83%–86%  B–: 80%–82%  C+: 77%–79%  C: 73%–76%  C–: 70%–72%  
D+: 67%–69%  D: 63%–66%  D–: 60%–62%  F: below 60%

- **Homework**  
  - Homework assignments will be based on your reading.  
  - You will have 2-3 homework assignments every week.

- **Project**  
  - The project will be completed by teams of two students  
  - The topic of the project will be the design and prototype implementation of an e-voting system

- **Quizzes** - There will be quizzes on the reading material

- **Comprehensive final exam**  
  - There will be a cumulative final exam.  
  - The exam will be closed book.  
  - You may use one cheat sheet, i.e., a regular sized page written in front and back.

**Late Submissions**

- Deadlines are firm.  
- You may submit up to two days late with 20% penalty for each day that you are late.  
- *A score of zero will be assigned to any project/homework that has not been submitted within two days after the deadline.*

**Re-grading**

If you have a request for re-grading, you need to ask me to re-grade your exam or homework up to one week (five business days) after this has been returned to you. *There will be no re-grading if the test/project that is older than one week.* I reserve the right to re-grade the full test/project. This means that I will not re-grade only the part you have requested, but the whole exam/homework and add or reduce points accordingly.

**Missed Exams**

If you miss an exam, the only way to take this test on a different day is to have an official document (ex.
from doctor, coach) verifying the reason you had to miss the test AND to let me know with an email BEFORE the exam, that you will miss the exam. Please refer to the student handbook “Class attendance policies” for a more detailed description of excused absences. A reason to miss the test may be a health issue, a sports tournament you had to participate, or an important personal issue. I will consider rescheduling on a case-by-case basis.

**Attendance**

Regular attendance is expected of all students. I take attendance at the beginning of each class session; however, I do not give you points for attendance. I do not take off points for not attending the class. For any grade to be awarded, participants must attend at least 85% of the class hours. Participants are expected to attend all sessions, *be punctual*, and remain for the duration of each class. In the rare case where some absence is required, make up work will be assigned where it is practical to do so.

Attendance is also part of the grading scale. Students may be withdrawn by the instructor if absences violate these guidelines.

**Schedule**

The schedule is tentative and *subject to change* during the semester.

<table>
<thead>
<tr>
<th>Week 1</th>
<th>Intro</th>
<th>ch.1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ch. 2</td>
</tr>
<tr>
<td>Week 2</td>
<td>UML</td>
<td>ch.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ch.2</td>
</tr>
<tr>
<td>Week 3</td>
<td>Legal considerations</td>
<td>ch.2</td>
</tr>
<tr>
<td></td>
<td>Communication</td>
<td>ch.2</td>
</tr>
<tr>
<td></td>
<td>ch.3</td>
<td></td>
</tr>
<tr>
<td>Week 4</td>
<td>Requirements</td>
<td>ch.4</td>
</tr>
<tr>
<td></td>
<td>ch.4</td>
<td></td>
</tr>
<tr>
<td>Week 5</td>
<td>Analysis</td>
<td>ch.5</td>
</tr>
<tr>
<td></td>
<td>ch.5</td>
<td></td>
</tr>
<tr>
<td>Week 6</td>
<td>System Architecture 1</td>
<td>ch.6</td>
</tr>
<tr>
<td></td>
<td>ch.6</td>
<td></td>
</tr>
<tr>
<td>Week 7</td>
<td>System Architecture 2</td>
<td>ch.7</td>
</tr>
<tr>
<td></td>
<td>ch.7</td>
<td></td>
</tr>
<tr>
<td>Week 6</td>
<td>OO Design 1</td>
<td>ch.8</td>
</tr>
<tr>
<td></td>
<td>ch.8</td>
<td></td>
</tr>
<tr>
<td>Week 7</td>
<td>OO Design 2</td>
<td>ch.9</td>
</tr>
</tbody>
</table>
Honor Code
I expect you to abide by the Honor Code and the Student Handbook: A Guide to Civil and Honorable Conduct. If you have a question about how to interpret the Honor Code, ask before acting! I encourage collaboration, but you must document it. Thus, each student will submit their own homework and, when collaborating, provide a reference to those people and documents consulted.

What is plagiarism?
The unauthorized use or close imitation of the language and thoughts of another author and the representation of them as one's own original work, as by not crediting the author. (Source: dictionary.com)
As you noticed above, I am citing the Internet source from which I used my information. Plagiarism includes using material from the Internet without citing the website from which you got your material.
Books, articles and any hard copy sources should be cited as well. **Plagiarism is considered cheating.**

**Plagiarism and coding (what you can and cannot do!):**

1. You may look up examples on the Internet.
2. You may **NOT** copy paste code from the Internet and present it as your own. Avoid copy pasting code from the internet and use this as a last resort ALWAYS with citation to the website that you used.
3. You may use libraries that are included in the Java API.
4. If you plan to use a library that is not on the Java API in a project, you will need to discuss this with me.

Discussing solutions with other students: Make sure you apply the **empty hand policy**, i.e., do not copy or use material from the discussion, just interact, brainstorm. You **cannot look at someone’s code and then type it. You cannot share the programs**, write code on a paper and share it with someone, or in any form whatsoever share your programs.

Collaboration in teams is allowed only if I have explicitly described in the project/homework assignment. You may collaborate based on the principles of pair programming (see below) and only if I have authorized teams. The Honor Code applies to the team members.

My actions after I suspect a cheating:

1. Contact the student and discuss the issue.
2. Consult with the honors committee and proceed to submit the issue with sufficient evidence that the student has cheated.

**Pair Programming**

Programming projects can be performed in teams of two members. The goal is to learn pair programming principles and extreme programming techniques that are used in industry. This allows the students to learn from each other and learn to collaborate. The main responsibilities for such collaboration are:

1. All the members of the team need to have project ownership, i.e., participate equally in the design, development and documentation. The instructor will ask in depth questions to all members of the team.
2. **All programming must be done in the pair.** Do not continue programming outside the pair. If you can’t finish in one session, meet again. If that’s impossible, save a copy of the code you pair-programmed for separate submission. Then work alone to finish the code. Review the part you coded alone with the other team members.
3. You need to follow the rules of pair programming, switching roles from observer to driver every 15 minutes or so.
4. All members receive the same grade.
5. A team leader will make the assignment submission. This is just to maintain one submission per team and in no way the team leader should do less or more work than the rest of the team members.
6. Students need to bring up collaboration issues early (first week of assignment) in order to switch teams.

**Accommodations for Adults with Disabilities**
The College will make reasonable accommodations for persons with documented disabilities. Students should apply for services at the Center for Disability Services/SNAP located on the first floor of the Lightsey Center, Suite 104. Students approved for accommodations are responsible for notifying me as soon as possible and for contacting me one week before accommodation is needed.

**Final Notes**
- I have a Greek accent that may be hard to understand sometimes. Please do not hesitate to ask me to repeat something.
- If you need to record the class, you may do this with your phone as long as you do not disturb the class.
- Please respect your classmates. Put your phone on silent mode before the lecture starts.