

COMP 2710-001/2 Software Construction Syllabus

Term: Spring 2019

Course: COMP 2710 Software Construction

Schedule: M/W/F, 9:00 – 9:50 am, Shelby 1103

Instructor: Dr. Xuechao Li

Office: 3101U Shelby Center

Email: xcl@auburn.edu

Office Hours: M/W/F 10 – 11:30am

GTA1: Zhitao Gong, zzg0009@auburn.edu

Office Hour: M.W.F. 10-11am

Office: Shelby 3139 desk #16 -----> changed to Shelby 2307 based on Zhitao announcement Jan 24th

GTA2: Zijie Zhang, zzz0092@auburn.edu

Office Hour: Wed 1:00pm - 4:00p----> changed to Mon 2pm-5pm based on Zijie announcement Jan 16th

Office: Shelby 3139 desk #16

Required Text: Savitch, Walter. *Absolute C++*, 6th Edition. Addison-Wesley.

You may also use: Savitch, Walter. *Absolute C++*, 2nd, 3rd, or 4th Edition. Addison-Wesley.

Reference: Dennis, Wixom, and Roth, *Systems Analysis and Design*. 5th Edition. Wiley.

Course Description

COMP 2710 is highly schizophrenic in that it is both a course on managing the complexity of large systems and an applied programming class. Managing software complexity requires some knowledge of software process. Applied programming means that you will be required to critically analyze real-world types of problems, design algorithms, and then implement those algorithms in high-level code to solve problems. COMP 2710 is as much about learning to solve problems as it is about C++ Programming. This course requires organization, effort, and discipline. You should prepare for every class and bring LOTS of questions – COMP 2710 is not a passive viewing experience. If at any time you feel that you are falling behind, you should contact the instructor immediately and come to office hours frequently. The keys to success in this course are attending every class, starting on homework assignments as soon as they are assigned, actively studying for exams, and always requesting help in a timely fashion.

This course typically requires 9 hours of time per week, on average for the average student. If you don't have it, drop.

Course Objectives

Upon successful completion of the course, the student should be able to:

1. Describe the difference between function-oriented and object-oriented software development
2. Analyze problems to determine system requirements
3. Develop object-oriented software designs that map to requirements identified in analysis

4. Grasp both C++ Syntax and Semantics
5. Develop software using sound programming principles
6. Use assertions and design by contract to develop high-reliability programs
7. Understand concepts of data abstraction, efficiency, and memory management
8. Understand how to perform unit and system-level testing
9. Apply a variety of strategies to debug simple programs
10. Use software tools in support of the software development
11. Demonstrate capability to use runtime configuration to bind variable values and program settings when a program is running
12. Construct, execute and debug programs on Linux

Three classes conducted by TA: March 18 – 22.

Assessment

Exams: Two Midterm Exams, Final Exam in room 1103 Shelby Center(2710-1) 1120 Shelby Center(2710-2)

Exams will be closed book, closed notes. Questions will be derived from lectures, material taught only in class, and from assignments. Question format will be mixed.

Midterm 1: 9 – 9:50am Feb 15th Midterm 2: 9 – 9:50am March 18th Final exam: 8 – 10:30am April 29th (2710-001)

Midterm 1: 1 – 1:50pm Feb 15th Midterm 2: 1 – 1:50pm March 18th Final exam: 12 noon – 2:30pm April 30th (2710-002)

Short Homework Assignments and Activities: 6 homework assignments

These activities will be take-home in nature and designed to reinforce concepts taught in class. They will be due in writing at the beginning of class. An electronic copy may also be necessary (specified in the assignment). Generally, these assignments are designed to be low-risk in the sense that they are designed to assess thinking and effort, rather than to strictly punish errors.

1. Learn how to use Linux and g++ (1 week)
2. Write a very simple C++ program (1 week)
3. Learn test drivers, assertions, random numbers (1 week)
4. Use arrays, streams, and file I/O (1 week)
5. Use structures, linked lists, multiple versions, and functions (1 week)
6. Learn how to define and implement classes, use strings and dynamic arrays (1 week)

Individual Construction Projects:

These projects will consist of the creation of design artifacts (turned in prior to the implementation) and correct C++ implementations of project specifications. All projects should be made to compile under the g++ compiler on Linux. **You may use any development platform or compiler, but your projects will be graded ONLY on a g++ compiler running on Linux. If your project does not work in that environment, you will NOT get credit. Always test it yourself in the lab (Shelby 2119 and Shelby 2122)!**

7. Project -Phase 1: Use tools (e.g., the argoUML tool) to create a data flow diagram and a use case diagram (1 week)
8. Project -Phase 2: Design and implement a non-trivial application using the function-oriented development approach, develop a singly linked list, gain experience with unit and system testing, and develop a reasonably user-friendly application (2 weeks)

Your work will be graded on Linux platform.

The instructor reserves the right to assess other penalties for any errors not strictly covered in the above rubric.

Grades:

- Exams 39% (Two midterm exams and one final exam)
 - Exam1 12%
 - Exam2 12%
 - Final Exam 15%
- Attendance 10%
- Assignments 42%
- Project 9%

A [90, 100], **B** [80,90), **C** [70,80), **D** [60,70), **F** [0,60)

Course Policies

Scaling, Curves, etc: Grades will not be scaled, curved, or adjusted arbitrarily. Do not expect a low-performance assignment/project/exam can be replaced by a high-performance one.

Project/Assignment Due Dates: Projects/assignment will be submitted through Canvas. Projects will always be due at 4:59 pm on the due date. Late assignments will receive a grade of zero (0). Deadlines will be made as generous as possible to *a priori* take into account illness, other courses, Acts of God, and nearly all conceivable excuses. If you have a documented illness preventing you from completing your assignment, you may submit all of your partial work and request an extension. **This extension is not automatic. No Late Submission.**

Email Policy: Your questions posted through emails are less likely to be answered (see the **Piazza** Section below), because the questions by emails cannot benefit other students. Instructor will answer homework and project questions on Piazza and Canvas. If a student asks a particularly relevant question, Instructor will post the response on **Piazza** in Canvas for the benefit of the entire class. You are responsible for all announcements made in class or electronically. You should read your Canvas information at least once every two days.

Piazza: We will adopt Piazza for class discussion. Piazza is highly catered to getting you help fast and efficiently from students, the TA, and Instructor. Rather than sending questions to me through emails, you should post your questions on Piazza.

Rebuttal Period: You will be given a period of three days to read and respond to the comments and grades of your project or assignment. The TA may use this opportunity to address any concern and question you have. The TA also may ask for additional information from you regarding your homework or project.

Classroom activities: The following activities are prohibited:

- alcohol drinks (i.e. martini, beer)
- headset
- video/video games/online video game with classmates(i.e. keyboard noise, quick movement of eyes, colors reflected from eyes glasses/pupil, quiet smile/leaning body to friends with digital devices, selfies)
- video dating
- snore and then whistles

Announcements: You are responsible for all announcements made in class or electronically. You should read your e-mail at least once a day. If a student asks a particularly relevant question, I may forward the response to the entire class for their benefit.

The instructor rejects to reply your emails if no course name/number and section number in the Subject line.

GTAs will reply your emails within 120 minutes in regular working days. 8am – 6pm Mon – Fri (except weekends/national holidays). It is at GTAs' discretions for the rest of time. For "Ping-pong" emails, if students do not reply GTAs' emails within 120 minutes, GTA will decide if he/she continues to communicate with students.

Special Accommodations: Students who need accommodations are asked to arrange a meeting with your instructor as soon as possible. If you have a conflict with my office hours, an alternate time can be arranged. To set up this meeting, please contact me by email. If you do not have an Accommodation Memo but need accommodations, make an appointment with The Program for Students with Disabilities, Office of Accessibility, 1228 Haley Center, 844-2096 (V/TDD).

<https://accessibility.auburn.edu/>

Please see the following link for more information on accommodations (modified 09/24/2018):

<https://sites.auburn.edu/admin/universitypolicies/Policies/ADAAccommodationsPolicy.pdf>

Students should proactively contact Instructor for a face to face meeting within 48 hours since students send a special request to Office of Accessibility. Otherwise, students should following regular rules without any exception.

Academic Integrity: Students will be expected to understand and follow Academic Honesty policies in place by the university. All work is to be done **individually**. Students should NOT share any project code or even detailed algorithm information with each other. Your programming code is exclusive to you.

Approved references:

The following constitute acceptable references to help you complete assignments.

- The course textbook is always approved and content may be used without citation.
- My course notes, lectures, and advice I give in my office may be used without citation
- Online general web references are fine, provided you give a citation for the website at the top of your code AND clearly label any lines of code that you use (it should never be ambiguous which lines of code you used from a website)
- Other books/textbooks on the language are fine, but require citations
- You are allowed to discuss broad conceptual ideas (for example, the idea of polymorphism)

with other students, but never to share code. If you discuss something with another student (even casually), you should always cite that reference in clear terms.

Unapproved references (these constitute Academic Dishonesty):

This is not a complete listing and cases of ambiguity should always be referred to the instructor for approval prior to use.

- Solution manuals for the text (or the like)
- Websites that sell custom code to individuals
- Code written by others (students or otherwise) for this class or similar classes
- Anything not listed under “Approved References” or approved by the instructor

You **MUST** document references clearly. If you discuss a project with another student or professor, you should indicate what you discussed and who you discussed it with clearly in the header of your project documentation (and/or code).

For example:

```
//Xuechao Li
//Project1.cpp
//Dr. Li helped me debug a syntax error in my “for” loop.
//I used Wikipedia.org in order to learn how a genetic algorithm works.
//I spoke with Bob Smith in the class about identifying objects in C++.
```

If you don’t need any sources for an assignment, clearly state “I did not use any external sources for this assignment” in your source code. Failing to document sources is plagiarism and will be penalized.

If you are unsure whether or not to document a source, it is better to document. Breaches of Academic Honesty will be referred to the Academic Honesty Committee and the strictest sanctions possible (including expulsion and failure) will be my recommendation.

If you are ever unclear about whether or not a course of action is unacceptable, you are always encouraged to consult the instructor.

Attendance: You are responsible for all materials and announcements presented in class (even if absent). Being late more than 15 minutes and/or leaving earlier than 15 minutes will be considered an absence.

- 1) For exams:
 - a. if you have a planned university-approved absence you must make me aware before the test in writing (with appropriate documentation).
 - b. if you have an unplanned absence, you must provide written, documented, and verifiable justification in the Exam date except Emergency such as coma.
- 2) If you are late for a test, you do not receive any extension.
- 3) Consistent attendance is typically essential to obtaining a good grade (C or better).
 - a. 0 point for your attendance if you are absent more than FOUR times without any proper excuses.
 - b. You will be reported to Department (Mr. Clint Lovelace) if you are absent consecutively more than FOUR times without any proper excuses.

- c. Calculation. (1) Sign-up sheets; and (2) Instructor's observation. You may not be informed for any absence since Instructor can memorize your name and face. Instructor does not recommend any student to challenge and/or test Instructor's memorizing ability.

Reading: Students are expected to read the appropriate sections of the book before each lecture.

Getting Help: Assignments may prove challenging and time-consuming. You are always welcome to bring questions concerning labs to the class, as well as to office hours. A good strategy is to always start early on projects, so that if you run into difficulties, you can get help as soon as possible. I will do my best to answer questions concerning labs within 24 hours of receiving them; however, I do not guarantee that I will always have time to debug code via e-mail (I prefer not to do so). For time-consuming problems dealing with code, office hours are always preferable. I will not help debug code via e-mail on the day an assignment is due.

Office Hours: You are always welcome to drop by during office hours to discuss projects or general concepts. To get urgent help or advice out of office hours, it is recommended to send an email in advance to make an appointment.

Course Difficulty: Typically, the course starts off relatively easy and gets harder as time goes on. Often, students are deceived by the (slower) initial pace and develop lazy habits at the beginning of the course. By mid-semester, they have thrown away many grade opportunities and find themselves in a bad situation with respect to grades. No amount of effort at the end of the class will compensate for consistent, dedicated effort throughout the class. Whether or not you have past experience with programming (or even with C++), my strongest recommendation is that you respect the class and come to class ready to engage every single class period. If you do this, you will dramatically increase your chances of success.

Bonus System (attendance).

1. In-class worksheets/quizzes. Appearance (0.1 point), correctness (0.1 point)
2. Random bonus (at Instructor discretion) will be distributed if a low-rate attendance happens someday.
3. Early-bird bonus.
 - Attendance 60.00%--69.99%: 0.4 points
 - Attendance 70.00%--79.99%: 0.7 points
 - Attendance 80.00%--84.99%: 1 point
 - Attendance 85.00%--100%: 1.5 points
4. Lottery bonus. (section 1: approved by 89/125=71.2%, 13 students absent on Jan 11, 2019
section 2: approved by 34/51=66.67%, 2 students absent on Jan 11, 2019)
 - It is only open to students who have at least 70% attendance.
 - For some (at Instructor discretion) exams, you may or may not be waived by 25%/40% job through a random name selector/black box game.
 - 25% job waived: 70.00%--84.99% [3% beneficiary—ceiling calculation]
 - 40% job waived: 85.00%--100% [2% beneficiary—ceiling calculation]
5. Donation

Students who get an A without any bonus can transfer bonus (100% or 0%) to students who desperately need a promotion and maintain at least 60% attendance. (each student can accept donation only once!)