

CSCE 1040 Syllabus

Instructor: Philip Sweany

Office: NTDP F262

Office Hours: Monday and Wednesday, 1:30 – 3:15; Tuesday, 1:45 – 3:45

Email: sweany@cs.unt.edu

Textbook:

Adam Hoover's

System Programming in C

Addison-Wesley, 2010; ISBN: 0-13-606712-03

Course Description:

CSCE 1040, the second course in the introductory sequence, focuses on more advanced C programming, designing and implementing larger software projects, introduction to dynamic data structures, and a brief introduction to C++ I/O and classes. The main focus is on developing students' software development skills.

Course Outcomes:

Course outcomes are measurable achievements to be accomplished by the completion of a course. These outcomes are evaluated as part of our ABET accreditation process.

1. Write readable, efficient, and correct C programs that include programming structures such as assignment statements, selection statements, loops, arrays, pointers, both standard library and user-defined functions, dynamic memory allocation and deallocation, any subset of C's rich set of operators and multiple header (.h) and code (.c) files.
2. Design and implement recursive algorithms in C.
3. Describe the concept of an abstract data type (ADT).
4. Use a combination of interactive and recursive design and implementation techniques to implement and use dynamically-allocated data structures in developing C applications.
5. Use a symbolic debugger to find and fix runtime and logical errors in C software.
6. Using a software process model, design and implement a significant software application in C. Significant software in this context means a software application containing at least five files, tens of functions and a makefile.
7. Implement, compile and run a small C++ program that includes a class definition and a main function to test the functionality of that class.

Policies:

- All programs are due at 11:59pm on the due date.
- No late programs will be graded
- All programs will be submitted through the Praktomat submission system.
- Except for the start of the term, attendance will not be taken in lecture. However, your attendance is strongly recommended to improve your opportunity to meet course outcomes

Grading: The components of your grade will be weighted as follows:

- 1 midterm exam, 20%
- 1 final exam, 25%
- Multi-week programming assignments, 30%
- Lab exercises and “minor assignments”, 25%

Make-Up Policy:

There will be no make-up exams, labs, or programs given in this class. However, for documented *excused absences* or *emergencies* on a day of an exam or a lab the exam and/or lab grades will be replaced by an average of the other exam or lab scores. There is one exception to this rule. Under NO circumstances will more than one exam or lab score be replaced by an average of the other scores. For a second missed exam or lab, even if all are excused, students will receive a 0 for the missed work.

Excused Absences

Students are expected to schedule routine appointments and activities so as not to conflict with attending class. However, some absences cannot be prevented. In the event of a medical *emergency* or *family* death, students must request an excused absence as quickly as feasible following the emergency. Use common sense. Students must provide documentation that verifies an emergency arose.

Emergencies

By definition, emergencies cannot be planned for. Your instructor attempts to make accommodations in these instances that allow for making up missed work and completion of the course in a timely manner. Among these emergencies are

- A death in your *immediate* family
- An accident or illness requiring immediate medical treatment and where a doctor has indicated attending class is impossible or inadvisable.
- Employees who are on call 24/7 fall in this category but must document that they were called during a scheduled class.

Collaboration and Cheating:

For minor assignments and labs (if any) I encourage collaboration among students (and faculty?) as I believe that it provides a better learning environment. And the

main goal of the minor assignments is to give students an opportunity to sharpen their software engineering capabilities. However, simply copying someone else's program solution will not help achieve that goal. And, it will, almost surely, cripple your ability to complete the major programs. A good criteria for identifying "helpful" collaboration is that you can design programs and even share programming tips in a group, but once you've done that and then moved on to something else for at least 24 hours, you should be able to write the program on your own.

However, on major programs, you should work alone. Do **NOT** work with other students on shared program solutions. Do **NOT** get help with algorithms or coding from anyone other than Dr. Sweany or the TAs. Do **NOT** use even partial program solutions from the internet. Failure to adhere to these strict standards will be cause for disciplinary action that could be as severe as expulsion from the university.

It **IS** permissible to obtain help from whoever you wish to fix **syntax errors**. And we will be discussing in class the different types of errors that occur in programs so there will be ample opportunity for you to learn the difference between syntax and other errors. But remember, for anything but syntax errors, getting programming assistance from any source other than Dr. Sweany or the TAs will be considered cheating and dealt with harshly.

And, of course you need to do your own work on quizzes and exams as well. Here there should be no ambiguity at all.

In case the above description, and in-class discussion of my views on appropriate and inappropriate collaboration does not answer all of your questions, please look at the university Student Rights and Responsibilities web page.

SETE

The Student Evaluation of Teaching Effectiveness (SETE) is a requirement for all organized classes at UNT. This short survey will be made available to you at the end of the semester, providing you a chance to comment on how this class is taught. I am very interested in the feedback I get from students, as I work to continually improve my teaching. I consider the SETE to be an important part of your participation in this class.

ADA

UNT complies with all federal and state laws and regulations regarding discrimination, including the Americans with Disability Act of 1990 (ADA). If you have a disability and need a reasonable accommodation for equal access to education or services please contact the Office of Disability Accommodation.

TENTATIVE Schedule and Topics

Week	Topic	Reading	Program
8/29/12	Introduction	Chapter 1	
9/10/12	Data Representation	Section 2.1	
9/17/12	Bitwise Operations	Sections 2.2-2.3	
9/24/12	Arrays and Strings	Sections 3.1-3.3	Major Pgm 1; 9/28/12
10/1/12	Command Line Args	Section 3.4	
10/8/12	Pointers	Sections 4.1-4.2	
10/15/12	Structures	Sections 4.3-4.4	
10/22/12	Midterm Exam		Major Pgm2; 10/26/12
10/29/12	Input/Output	Sections 5.1-5.5	
11/5/12	Program Organization	Sections 6.1-6.2	
11/12/12	Makefiles	Sections 6.1, 6.3	Major Pgm3; 11/16/12
11/19/12	Lists	Online Material	
11/26/12	Lists, G++	Online Material	
12/3/12	G++ Classes	Online Material	Major Pgm4; 12/6/12
12/10/12	Finals		