CSCE 1030 - Computer Science I

Instructor: David Keathly
Office: NTDP F201J
Office Hours: Tuesday, Thursday 9:00am-11:00am

Email: david.keathly@unt.edu

Semester: Spring 2014
Place NTDP B185
Section 1: MWF 8:30 – 9:20 am
Section 2: MWF 10:30 – 11:20 am

Additional Resources:
TA support: F270
Peer tutors
Class website

Course Catalog Description

CSCE 1030 is the introductory course for the computer science, computer engineering and information technology degrees offered by the Department of Computer Science and Engineering. As such it introduces students to the broad discipline of computing while placing emphasis on developing students' programming skills. In addition to two 75-minute "lecture" classes per week, each student will participate in a three-hour laboratory session each week.

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. Describe how a computer's CPU, Main Memory, Secondary Storage and I/O work together to execute a computer program.
2. Make use of a computer system's hardware, editor(s), Linux operating system, system software and network to build computer software and submit that software for grading.
3. Describe algorithms to perform "simple" tasks such as numeric computation, searching and sorting, choosing among several options, string manipulation, and use of pseudo-random numbers in simulation of such tasks as rolling dice.
4. 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, and multiple header (.h) and code (.c) files.
5. Use commonly accepted practices and tools to find and fix runtime and logical errors in C software.
6. Describe a software process model that can be used to develop significant applications composed of hundreds of C functions.
7. Perform the steps necessary to edit, compile, link and execute C software.

Textbook:

Kelly and Pohl, A BOOK ON C (4th edition) Addison Wesley, ISBN 0-201-18399-4

Prerequisites

Completion or concurrent enrollment in MATH 1650

Course Requirements:

Attendance:	Optional, although student is responsible for all materials covered in lecture and class discussion
Exams:	Three
Assignments:	There will be programming assignments, quizzes, exams, and laboratory exercises to complete

For More information

Class Web Page:	http://moodle.cse.unt.edu
Faculty Web Page	www.cse.unt.edu/~dkeathly

Course Calendar (subject to change)

Week	Topics	Readings and Materials
Week 1	Class Overview Intro to Problem Solving, Programming, and C	Online resources Chapter 0
Week 2	Data types, variables, assignments and operators Lab 1	Chapters 2 and 3
Week 3	Input and output Lab 2, Quiz 1	Chapter 11.1 – 11.3
Week 4	Flow Control Lab 3, Quiz 2	Chapter 4
Week 5	Flow Control Lab 4, Quiz 3	Chapter 4
Week 6	Strings Lab 5, Quiz 4	Chapter 6.10 – 6.11
Week 7	Functions Lab 6, Quiz 5	Chapter 5
Week 8	Exam 1 Lab 7, Quiz 6	
Week 9	Arrays Lab 8, Quiz 7	Chapter 6.1, 6.12
Week 10	Pointers Lab 9, Quiz 8	Chapter 6.2 – 6.8
Week 11	Standard Libraries	Appendix A

	Lab 10, Quiz 9	
Week 12	Exam 2 Large Program Development Lab 11, Quiz 10	Online resources
Week 13	Algorithms Lab 12, Quiz 11	Online resources
Week 14	Random Numbers Lab 13, Quiz 12	Online resources
Week 15	Additional topics TBD	Online resources
Week 16	Final Exam Week (Exam 3)	

Grading Policy

The various components of your grade are weighted as follows:

Lab Programs (13 drop 1)	15%
Quizzes (12 drop 1)	10%
Programming Assignments (6-7)	30%
Exams (3)	45%
(3 exams 15% each)	

Calculators will not be needed or allowed. Make up exams will only be allowed under exceptional circumstances (e.g., a note from your doctor).

Final grades in this class will be determined based on the following scale: A: over 90%, B: 80% - 89%, C: 70% - 79%, D: 60% - 69%, F: below 60%.

Note: You must pass BOTH the homework portion of the class (lab programs, quizzes, and programming assignments) AND the exam portion (midterms and final) with a grade of D or better in order to pass this course. Hence, an overall average greater than 60% may still result in a failure in some cases. Note that while a D passes the class, a grade of C or better is required for graduation.

Course Policies:

- Academic Misconduct:

  The department, college, and university have very strict guidelines regarding academic misconduct. Copying is not allowed on exams. Students are expected to submit their own work on individual programming projects (labs and homework). Do NOT work with other students on shared program solutions. Do NOT get help with algorithms or coding from anyone other than your instructor or the lab TAs. Do NOT use even partial program solutions from the internet. *We*

will use a very sophisticated program to compare your work to the work of all other students (including students in past classes if appropriate). If it is determined that you have cheated, the first instance of cheating in the class will result in a grade of ZERO on the project or exam in question and referral to the department chairman and dean of engineering. The second instance of cheating in the class will result in a grade of F in the class, and a dismissal hearing may be initiated by the dean of engineering.

- 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 your instructor or the Class 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.
- You are responsible for the information covered in class, whether you attend class or not. Individualized lectures will not be given. Please check with other class members for any notes that might have been missed during an absence. 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 objectives.
- Students should expect an "in-lab" program each week in lab. The program will be submitted before that lab session is complete.
- Weekly quizzes will be completed online via the class webpage.
- All non-lab programming assignments are due at 11:59pm on the due date. **Programming assignments will be accepted up to 24 hours late and late programming assignments will be assessed a 50% grade reduction penalty. After 24 (exactly!) hours, late programming assignments will receive a grade of zero.** Partial credit will be given for programs which compile but which are not complete. Starting early on programming projects is strongly encouraged. Students typically have great difficulty completing their projects in one night the day before they are due. Students are allowed to discuss program design and other high level issues with each other. Students are also allowed to help each other understand specific compiler or run time error messages. Copying all or part of another person's program is strictly prohibited and will result in a grade of zero. Supplying printed or electronic copies of your homework to other classmates will also result in a grade of zero. All programs will be submitted through the class website.
- All pertinent information about the class (assignments, exam reviews, sample code, written topic discussions, and day-to-day event schedule) are available via the class webpage. If there is ever a question as to when something is due, or an additional copy of a course document is needed, ALWAYS check the class webpage. If you feel there is incorrect or there is missing information on the class website, email the instructor about the problem immediately. Electronic mail (email) will be a major means of

communication with the instructor outside of actual classroom discussions.
- Please keep this information sheet handy during the semester and always periodically check the class homepage for any course information, including scheduling of programming assignments, exams, and exam reviews.

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, etc., students must request an excused absence as quickly as feasible following the event. Use common sense. Students must provide documentation that verifies the reasoning for the excused absence.

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. Students must provide documentation that verifies the emergency.

Student Evaluation of Teaching Effectiveness (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 and notify your instructor.