CSCE 1030 Computer Science I

Instructor: David Keathly  
Semester: Fall 2016  
Office: NTDP F202  
Lecture MW 3:30 – 4:50 ESSC 255  
Office Hours: MW 10:00 – 12:00 and TTh 10:00 – 11:30  
Phone: 940-565-4801  
Email: david.keathly@unt.edu (email will get better response than phone)

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:
Walter Savitch, Problem Solving with C++, 9th Edition, Addison-Wesley

Prerequisites
Completion of CSCE 1030 with a grade of C or better.

Course Requirements:
Attendance: Highly recommended as student is responsible for all materials covered in lecture and class discussion
Exams: Two
Assignments: There will be some larger programming assignments, quizzes, exams and laboratory exercises to complete

For More information
Faculty Webpage: Faculty.unt.edu
Class Web Page: Blackboard Learn

Course Plan
My plan this semester is that we will use half the class period for traditional lectures and the second half as problem solving, homework help, etc. Some lectures may only be recorded and you will have TA and grader help during class time when I am traveling (about 2 weeks out of the Fall session). This means you should be able to begin working on homework early and get most of it done during class time. You can also get additional help during office hours and during lab after you complete the lab assignments.
<table>
<thead>
<tr>
<th>Week</th>
<th>Topics</th>
<th>Readings and Materials</th>
</tr>
</thead>
</table>
| Week 1| Course Overview
Intro to Problem Solving, Programming and C/C++
Basic CPU Architecture
Data Types | Chapter 1 |
| Week 2| Compiler Understanding and Operation
Program Structure
Comments
Quiz 1
Lab 1 (using tools) | Chapter 1 |
| Week 3| Algorithms
Expressions, Statements and Operators
Lab 2 (simple Programs and Compiling)
Quiz 2 | Chapter 2 |
| Week 4| Arithmetic and Logical Expressions
Lab 2 (Variables and Expressions)
Quiz 3 | Chapter 2 |
| Week 5| Loops, Conditionals and Flow Control
Lab 3 (Flow Control)
Quiz 4 | Chapter 2 and Chapter 3 |
| Week 6| Flow Control
Enumerated types
Lab 5 (enumerated types, switches, BOOL)
Quiz 5 | Chapter 3 |
| Week 7| Functions
Lab 6 (Using Functions)
Quiz 6 | Chapter 4 |
| Week 8| Arrays
Lab 7 (Writing Functions)
Quiz 7
Exam 1 (covers material through Week 6) | Chapter 4 |
| Week 9| Console I/O (C and C++ styles)
Pointers
Lab 8 (Functions calling Functions, Debugging)
Quiz 8 | Chapter 6, Chapter 9 |
| Week 10| File I/O (both styles)
Lab 9 (File I/O) | Chapter 6 |
| Week 11 | Lab 10 (Arrays)  
Quiz 10 | Chapter 7 |
|---------|----------------|------------|
| Week 12 | Strings (both Types)  
Lab 11 (Arrays and Strings)  
Quiz 11 | Chapter 8 |
| Week 13 | Structures and Unions  
Command Line Arguments  
Lab 12 (Pointers and Arrays) | Chapter 10.1 |
| Week 14 | Debugging  
Using Libraries | Chapter 1 and Chapter 4 |
| Week 15 | Misc Topics  
Exam Review | |
| Week 16 | Final Exam (Exam 2) | |

**Grading Policy**

The various components of your grade are weighted as follows:

Lab Programs (12 drop 1) 30%
Quizzes (11 drop 1) 10%
Larger Programming Assignments (6 – no drops) 40%
Exams (2, 10% each – no drops) 20%

**Course Policies:**

- On programs, do your own work. 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 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. 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.
• 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.

• There will be no make-up exams, quizzes, or programs given in this class. However, for documented *excused absences* or *emergencies* additional time for homework or an alternate lab date may be granted. Exam makeups or substitutions may be granted as well depending on the situation. Note these exceptions are only in the case of documented excused absences or emergencies. In most cases you should contact the instructor before the absence to make proper arrangements.

• 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.

• Weekly quizzes will be completed online via the class webpage.

• Students should expect an "in-lab" program each week in lab. The program will be submitted before that lab session is complete. You must make arrangements in advance if you are going to miss your assigned lab section. All labs must be completed within the calendar week they are assigned. All work will be completed in lab unless otherwise instructed by your lab TA.

• There is no curve grading in this class. However the instructor does maintain a “fuzzy borders” policy at the end of the semester for students who complete every lab, homework assignment, quiz and exam. This means that grades that are close to a border (e.g. 87.5 – 89.4) might round up to the next higher grade if students have completed all assignments and have maintained good performance on homework and labs, but perhaps fallen a bit on quizzes or exams. Details of this policy will be discussed during the first class period.

• 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 only be given for programs which compile but which are not complete. No regular or late credit will be given for programs that do not compile! 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.

• The instructor, peer mentors and TAs require a current copy of the program when a student is asking a question about a program.

• All pertinent information about the class (assignments, exam reviews, sample code, written topic discussions, and day-to-day event schedule) is 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, 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.

**Student Evaluation of Teaching Effectiveness**

The Student Evaluation of Teaching Effectiveness 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 your evaluations 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.