

CSCE 1040 Computer Science II

Instructor: David Keathly Semester: Spring 2014

Office: NTDP F201J

– 12:20 pm

Office Hours: TTh 9:00 am – 11:00 am

Phone: 940-565-4801

Email: david.keathly@unt.edu

Times: Section 001 TTh 11:30 am

Place: NTDP B185 Lab F270

Course Catalog 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 with at least five files, ten 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.

Textbook:

Kelly and Pohl, A BOOK ON C (4th edition) Addison Wesley, ISBN 0-201-18399-4 (will be textbook in Fall 2013)

Prerequisites

Completion of CSCE 1030

Course Requirements:

Attendance: Optional, although 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: www.cse.unt.edu/~dkeathly

Class Web Page: <http://moodle.cse.unt.edu>

Course Calendar (subject to change)

Week	Topics	Readings and Materials
Week 1	Class Overview Programming Review	Chapters 1-5
Week 2	Arrays and Strings Command Line Args File I/O Lab 1 (Linux)	Chapters 6 and 11
Week 3	Recursion Quiz 1, Lab 2 (Printing) Hwk 1 due	Chapter 5.14 – 5.15
Week 4	Hash Tables Lab 3 (Recursion), Quiz 2	handouts
Week 5	Pointers and Strings Quiz 3, Lab 4 (Hash/Search)	Chapter 6
Week 6	Bits, Bytes and Operators Lab 5 (Pointers), Quiz 4 Hwk 2 due	Chapter 7
Week 7	Structures and Unions Lab 6 (Bit/Byte Manip), Quiz 5	Chapter 9
Week 8	Midterm	
Week 9	Developing Large Programs	Chapter 5 Section 5.8-5.12

	Lab 7 (Lab Exam), Quiz 6 Hwk 3 due	
Week 10	Stacks and Queues Lab 8 (Structures), Quiz7	Chapter 10
Week 11	Lists and Trees Lab 9 (Debugging/Lg. Prgm), Quiz 8	Chapter 10
Week 12	Objects and Classes / OOAD Lab 10 (Lists), Quiz 9 Hwk 4 due	Chapter 13 and handouts
Week 13	Intro to C++ Lab 11 (C++ 1) , Quiz 10	Chapter 13 and handouts
Week 14	I/O in the Brave New World Lab 12 (C++ 2), Quiz 11	Chapter 13 and handouts
Week 15	Additional Topics TBD Hwk 5 due	handouts
Week 16	Final Exams (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 (5-6)	30%
Exams (2, 15% each)	30%

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. 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.
- There will be no make-up exams, quizzes, 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 exams or lab scores. There is one exception to this rule. Under NO circumstances will more than one exam or two days worth of lab scores be replaced by an average of the other scores. For a second missed exam or third missed lab, even if all are excused, students will receive a 0 for any work that day.
- 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 competed 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.
- 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.
- The instructor 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) 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, 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 (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.