CSCE 1040 Computer Science II

Instructor: David Keathly
Office: NTDP F202
Phone: 940-565-4801
Place: see Canvas course page
Time: see Canvas course page
Location: see Canvas course page
Office Hours: See Canvas course page and by appointment

Email: david.keathly@unt.edu

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 beginning exploration of Object Oriented paradigms using C++ . 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 for all programming constructs defined for Programming Fundamentals I plus dynamic memory allocation, bit manipulation operators, exceptions, classes and inheritance.
2. Design and implement recursive algorithms in C/C++.
3. Use common data structures and techniques such as stacks, queues, linked lists, trees and hashing
4. Create programs using the Standard Template Library.
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 make file.
7. Implement, compile and run C++ programs that includes classes, inheritance, virtual functions, function overloading and overriding, as well as other aspects of Polymorphism.

Expected Student Outcomes:

Student Outcomes are measurable achievements to be accomplished by the completion of the degree. These outcomes are evaluated as part of our ABET accreditation process.

Computer Engineering Students:

1. An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
2. An ability to acquire and apply new knowledge as needed, using appropriate learning strategies

Computer Science Students:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
3. Apply computer science theory and software development fundamentals to produce computing-based solutions.

Information Technology Students:

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

Textbook:

We are using an E-Book this semester as the required textbook
Instructions for purchasing and using the textbook will be posted on Canvas, or a code can be purchased at the University Bookstore

Recommended Reference Book

Walter Savitch, Problem Solving with C++, 9th Edition, Addison-Wesley
ISBN-10: 0133591743 **ISBN-13:** 978-0133591743

Prerequisites

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

Course Requirements:

Attendance: Required - student is responsible for all materials covered in lecture and class discussion and there will be occasional quizzes in class
Exams: One Final and 4 Assessment Exams
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:	Canvas

Course Plan

You will have readings and small participation and Challenge Assignments that you must complete before coming to each class period. In class we will review the material for the day and any questions. At least one class period each week we will focus on developing some code in class, usually related to your homework or lab assignments. We will also be doing some special activities related to broadening your critical thinking, Problem solving and communications skills as it relates to a professional workplace.

The detailed calendar will be available in Canvas

Grading Policy

The various components of your grade are weighted as follows:

Lab Programs	15% drop 1
Participation Assignments (Zybooks and in class quizzes/activities)	10% drop 1
Challenge Assignments (ZyBooks)	10% drop 1
Programming Assignments Homework	20%
Final Exam	5%
Assessment Tests in Lab (4)	40%

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 Textbooks, the Internet or other sources, particularly without proper citation. Failure to adhere to these strict standards will be cause for disciplinary action that could be as severe as failure in the course or expulsion from the university. Using outsourcing applications and websites to have others complete your homework will result in an immediate failing grade in the course.

- It IS permissible to obtain help from fellow student, as well as the instructor, graders, TA's and Peer Mentors 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, Graders, Peer Mentors or the Class TAs will be considered cheating and dealt with harshly.

- You need to do your own work on participation and challenge assignments and exams as well. Here there should be no ambiguity at all.

- This class will use a single strike policy in that one instance of serious cheating or other form of academic dishonesty will result in a failing grade for the Course, less serious infractions will result in a zero grade for the assignment. The infraction will also be reported to the office of Student Affairs. This office maintains a database that is available to other instructors as well as employers. This could seriously affect your future employability.
-
- 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.

- Students should expect an "in-lab" program each lab period. 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.

- All non-lab programming assignments are due at 11:59pm on the due date. **Programming assignments will not be accepted late beyond a 30 minute grace window unless there are verifiable system problems or outages of service on Canvas.** 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 on the CSE Linux servers or in the E-Book environment as assigned! The only points you may earn for non-compiling assignments are any points for design or report components.** 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 Canvas website. No assignments will be accepted for grading via email or on paper.

- The instructor, peer mentors and TAs require a current copy of the program when a student is asking a question about a program. Questions must be specific. Please do not send a program and simply say "This does not work". Also please be prepared to show and explain your design when asking questions. If you have not completed a design then the TA, Peer Mentor or Instructor may require you to complete one before answering questions about program logic for your code.

- 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 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.

SPOT

The Student Perception of Teaching (SPOT) survey is a requirement for all organized undergraduate 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 SPOT 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.