CS 2308: Foundations of Computer Science II
Summer 2023
Section 501

Instructor: Gentry Atkinson (he/him)
gma23@txstate.edu
Derrick M14

Virtual Office Hours: (TBD)
Zoom Links are posted on Canvas

Important Dates:

May 29th	Memorial Day
May 30th	First day of Class
May 31st	Last day to add a class
June 2nd	Last day to drop with no record
June 23rd	Deadline to apply for graduation
June 30th	Final Exam, 11:00 AM – 1:30 PM

These dates may change.

Class Meetings: MTWRF 10:00-11:40 AM in Derrick 234

Course Description:
Fundamentals of object-oriented programming. Introduction to abstract data types (ADTs) including lists, stacks, and queues. Searching and sorting. Pointers and dynamic memory allocation. A continuation of CS1428. Prerequisite: "C" or higher in CS 1428.

Course Format:
- This class will held **in-person** but some lectures may be posted as videos in the event of time conflicts or to provide supplemental content.
- Exams, quizzes, and assignments will be delivered through the Canvas LMS. Students will need access to reliable internet to complete homework assignments.

Required Materials:

- Textbook- **Tony Gaddis, Starting out with C++: From Control Structures through Objects**, 10th Edition, ISBN: 9780137450626 (any edition can be used!).
- Students will need access to a computer with a web browser and an Integrated Development Environment (IDE). All class coding demonstrations will be done using the **CodeBlocks** IDE. Students may use any IDE for their own work but are responsible for installing and configuring that software for themselves.
- Students are encouraged but not required to bring a laptop or tablet to class for participating in live coding. **Bringing a laptop or tablet to class is required on exam days.**

Course objectives:

At the end of the course, the students should be able to:

- Describe and demonstrate at least two different algorithms for searching and at least two different algorithms for sorting.
- Implement a divide-and-conquer algorithm to solve an appropriate problem (binary search).
- State the time/space efficiency of various algorithms (using one of 6 categories of mathematical functions).
- List the 6 categories of mathematical functions used in analyzing algorithms in order from slowest to fastest growing.
- Read and write C++ code that uses pointer variables and memory operations (new, &, *, delete), including pointers to arrays, structures, and objects and the -> operator.
- Write C++ code that resizes an array using dynamic memory allocation.
- Write C++ code that deletes dynamically allocated memory to avoid memory leaks.
- Describe the basic concepts of object-oriented programming.
- Design, implement, test, and debug simple programs (using objects) in an object-oriented programming language (C++).
- Describe how the class mechanism supports encapsulation and information hiding.
- Develop (implement) programs using multiple classes and arrays of objects
- Develop and use appropriate algorithms, especially for processing lists (insert, remove, search, sort, etc.)
- Describe structured programming in terms of modules and functions.
- Develop (implement) programs with source code separated into multiple files, including header (.h) files
- Create, compile, and run a C++ program in a Unix style command-line environment
- Develop (Implement) C++ programs that create and use simple linked-lists, including code to insert into, delete from, and traverse a linked list structure.
- Compare and contrast the costs and benefits of dynamic and static data structure implementations.
- Describe the principle of the Abstract Data Type (ADT) and, in particular, explain the benefits of separation of interface and implementation.
- Implement user-defined data structures in a high-level language.
- Implement the list, stack, and queue ADT using arrays and linked lists.
- Write programs that use each of these data structures: linked lists, stacks, and queues.

Attendance Policy:
Students are encouraged but not required to attend the twice-weekly lectures. Lecture slides will be posted on Canvas, but not videos of live lectures. Students who choose not to attend lectures are still expected to learn all of the material presented in this class and may have difficulty completing assignments.

Some lectures may be moved to an asynchronous video format in the event of emergency or to accommodate the instructor's travel schedule. No more than four lectures will be delivered by video this semester.

Grading:

Pre-term exam	5%
Midterm exam	15%
Final exam	20%
Coding Projects (4)	40%
In-Class Assignments	10%
Independent Quizzes	10%

Exams:
The pre-term exam will test students' knowledge of material covered in CS1428. Students should take this exam on their own time following the first lecture. There will be a supplementary assignment to allow students to raise their grades on this exam if they wish to do so.

The midterm and final will be given through Canvas. If possible these exams will be given in-person. Students should arrange to have access to a laptop or tablet that they can bring to class on exam days.

Specific instructions for taking exams will be posted on Canvas.

A "curve" may be added to the midterm and final exams, but is never guaranteed. When a curve is added, it will be computed using the formula: (100 – the highest grade in the class).

Coding Projects:
There will be four coding projects throughout the semester. Students will implement a piece of software following a software design document posted on Canvas. These are individual assignments. Students may discuss the assignment with each other, but everyone should cite their collaborators and submit their own copy.

Submitting a solution that has been posted online will earn a 0 on that project.

Weekly Assignments:
There will be a small, weekly coding challenge posted on Canvas. These challenges will focus on re-enforcing the concepts discussed in lecture during that week of class as well as giving opportunities to practice clean coding practices. These assignments are collaborative, but students must turn in their own copy of the code they have written.

We will work on these assignments together in class every week. Students are encouraged to bring a laptop or tablet to work on these assignments during class time.

Late and Make-up Work:
Make-up exams will *only* be given for students who arrange an absence ahead of time or who are given an excused absence by the Dean of Students. Students who know that they are not going to be available during an exam time should contact the instructor to arrange an alternative testing time as soon as they find out about the conflict.

Students who experience illness or personal tragedy that prevents them from submitting work should contact the Dean of Students to request an excused absence. Students who are given an excused absence by the Dean do not need to explain the nature of the illness or tragedy to the instructor and will be given an opportunity to submit any work or exam that was due during the excused period.

The four Coding Projects each have a 24 hour period for late submissions. Projects submitted during the late submission period will be penalized by 20%.

Students are always welcome to request additional working time on an assignment or quiz (other than the midterm and final exams) through email, but these will be considered on a case-by-case basis.

Technical Difficulties :
If a failure of Canvas or the school's network interferes with an exam, a poll will be posted on Canvas to determine a "mitigation policy" for the grades on that exam. Options will include but are not limited to dropping a portion of the exam, applying a uniform curve to all exams, or re-taking the exam.

Network or software failures will not be considered when grading the Weekly Assignments, Coding Projects, and Independent Quizzes. Students responsible for finding access to a stable internet connection and for knowing how to use the IDE the have chosen to install. Computers with a wired internet connection and a working IDE maintained by the CS department are available in Derrick 231 and MCS 590.

Class Conduct:
Students should conduct themselves in a way that does not interfere with the learning of other students. This includes not using offensive or disruptive language, not interrupting lectures, and not sharing offensive or disturbing material through online forums. This policy applies whether our class meeting are virtual or in-person.

Disruptive conduct in class will receive a verbal or written warning from the instructor. Students who repeatedly disrupt class will be referred to the Dean of Students.

Students may bring well behaved children or infants to class with them but are responsible for their child's behavior and will be asked to remove the child from the classroom if they become disruptive.

Communication Policy:
Students that need to contact the instructor outside of class and office hours should do so through email using the address at the top of this syllabus. I do not monitor and will not respond to communications through other channels (e.g. Canvas, Slack, Discord, Facebook). I will try to respond to emails that I receive Monday-Friday within 24 hours or within 48 hours for emails that I receive on Saturday or Sunday. Students should be aware that my responses might be slower before major deadlines.

Withdrawals/drops:
You must follow the withdrawal and drop policy set up by the University:
http://www.registrar.txstate.edu/registration/dropping-or-withdrawing.html

Academic Honesty:
Students should remember that cheating only devalues their own education. You are expected to adhere to the University's Academic Honor Code:
https://www.txstate.edu/honorcodecouncil/Academic-Integrity.html

- You may work together on your programming assignments. If you submit a program that is the result of group work, you must list the names of all contributors in the file comments. Each student must submit their own program, even if it is the same as another students'. The penalty for not citing collaborators will be -30 points for that assignment.
- Do not include code obtained from the internet or any other source in your programming assignment (except what is provided by the instructor during the current semester). Do not post your solution anywhere on the internet. The penalty for either of these violations will be a 0 for that assignment.
- Submitting work done by others as if it were your own is an act of dishonesty.
- The penalty for violating academic honesty on a test or exam is a reduction of points up to a grade of 0 on the test or exam.
- Students who engage in dishonest activities will have to explain their actions to the Honor Code Council.

Accommodations for students with disability:
Any student with needs requiring special accommodations should contact the Office of Disability Services(ODS) at the LBJ student center (5-5.1). Students who qualify for extra time for exams must take their test with ATSD and must schedule their test at the same time the test is given in class. Note: you must submit your request with ATSD at least 2 business days before the exam date!

Students are never required or expected to discuss their disability with an instructor.

Course Schedule:

Topic	Dates
Unit 0 Class Introduction	Week 0 ()
Unit 1 Functions, Arrays & Structs	Week 1 () Week 2 ()
Unit 2 Searching, Sorting & Analysis	Week 4 () Week 5 ()
Unit 3 Pointers & Dynamic Memory Allocation	Week 6 () Week 7 ()
Review and Midterm	Week 8 ()
Spring Break	Week 9 ()
Unit 4 Introduction to Classes	Week 10 () Week 11 ()
Unit 5 Linked Lists	Week 12 () Week 11 ()
Unit 6 Stacks & Queues	Week 12 () Week 13 ()
Review	Week 14 ()
Final Exam	Week 15 ()

Extra Credit:
Five extra credit problems are offered during this course. They will be selected by the instructor from the coding challenge website HackerRank. Instructions for completing these problems will be posted on Canvas. Each problem completed satisfactorily will be worth up to one percentage point on a student's final grade.

No other extra credit is offered in this class.

This document may be updated later to correct typos, clarify statements, or to adjust the schedule.