

CS 3520 - Programming in C++

Course Overview

Examines how to program in C++ in a robust and safe manner. Reviews basics, including scoping, typing, and primitive data structures. Discusses data types (primitive, array, structure, class, string); addressing/parameter mechanisms (value, pointer, reference); stacks; queues; linked lists; binary trees; hash tables; and the design of classes and class inheritance, emphasizing single inheritance. Considers the instantiation of objects, the trade-offs of stack vs. heap allocation, and the design of constructors and destructors. Emphasizes the need for a strategy for dynamic memory management. Addresses function and operator overloading; templates, the Standard Template Library (STL), and the STL components (containers, generic algorithms, iterators, adaptors, allocators, function objects); streams; exception handling; and system calls for processes and threads.

Course Section

MWR 9:15 - 10:20 am - Richards 254

Instructional Team

Professor:

Mark Fontenot, PhD m.fontenot@northeastern.edu

Office: Meserve Hall 353

Office Hours: Mon & Thursday 12:00 - 1:30pm

(There's no way to choose office hours that work for everyone. If the above doesn't work for you, always feel free to reach out via Slack and we can schedule a time to chat.)

Teaching Assistants:

- Abhishek Kumar - kumar.abhishe@northeastern.edu
- Tarun Malviya - malviya.t@northeastern.edu
- Dhavan Sanghvi - sanghvi.dh@northeastern.edu

Questions and Communication

Join Piazza: You'll be added and receive info on joining/signing up.

Join Slack: Follow [this](#) link if you're not already part of Fontenot's Class Slack Org

- Use your Northeastern Email
- Join the #23f-cs3520 channel

Learning Outcomes

1. Learn how to program using C++ in a clear, efficient, and safe manner
2. Understand simple and complex data structures in C++.
3. Learn about memory management (allocation and deallocation) manually and automatically
4. Explore some of the built-in libraries of the STL

5. Expand debugging abilities using various tools
6. Learn how to structure larger C++ programs and use a build system such as CMake

Textbook

There is no REQUIRED textbook for this class. However, you may find the following helpful throughout the course

- Horton & Van Weert. *Beginning C++ 17 - From Novice to Professional*. Apress. ISBN: 978-1-4842-3365-8
- Gregoire. *Professional C++, 4th Edition*. Wrox Press, 2018. ISBN: 978-1-119-42130-6

These books and many others are available through O'Reilly Publisher's Higher Education Portal which is available the University Library Website. Go [here](#) and create an account with your Northeastern email. I've created a playlist (like a bookshelf) specifically for this class which you can access [here](#) after you've made your account. My personal and much more expansive C++ Playlist on O'Reilly can be found [here](#) (don't worry, we aren't going to cover all of these in this course). As the semester progresses, I may add other resources both inside and out of the O'Reilly portal.

Evaluation

The relative weights of the various assessment types is given below:

- In class activities: 10%
 - Cannot be made up. 1 or more will be dropped at the end of the term.
- Homeworks: 20%
 - Cannot be made up. 1 or more will be dropped at the end of the term.
- Programming Assignments: 50% (evenly weighted)
- Final Project: 20%

Grading Scale:

- | | | | |
|------|----------|------|----------|
| • A | 93 - 100 | • C | 73 - <77 |
| • A- | 90 - <93 | • C- | 70 - <73 |
| • B+ | 87 - <90 | • D+ | 67 - <70 |
| • B | 83 - <87 | • D | 63 - <67 |
| • B- | 80 - <83 | • D- | 60 - <63 |
| • C+ | 77 - <80 | • F | <60 |

Final grades will not be rounded at the end of the semester.

Programming Assignments:

All programming projects will be posted on the class webpage. Submission details will be contained within the assignment itself. Submission will happen via Github. No programming assignments will be accepted by means other than what is indicated in the assignment (not accepted via email, slack, etc.). In order to receive any credit at all, your program must at least build/compile. If your program does not build/compile, the score is automatically a zero.

Rather than penalize late submissions, I prefer to incentivize getting started and finishing early. For Programming Assignments, you can earn 3% bonus points on each one that is 48 hours BEFORE the stated deadline. *Note this is not available on Homework assignments or the final project!*

In recognition of “life happens”, everyone gets **one free 48 hour extension, no questions asked** on one homework assignment. This cannot be used on any course project deliverables. You just need to let Dr. Fontenot know if/when you’d like to take advantage of this. Other than this, **no other late work will be accepted.**

Homework Assignments: (added Sept 9, 2023)

These will be written problem sets that will be submitted to Gradescope. No late submissions will be accepted. In lieu of accepting late submissions, I will drop one homework grade at the end of the semester.

Project:

There will be a team project later part of the semester. It will be an opportunity for you to develop a larger C++ project. More information on the project will be released later in the semester.

Exams:

There will be no exams in the class.

Academic Conduct and Integrity

Submitting work that is not your own is **wrong**. Facilitating someone else in submitting work that is not their own is **wrong**. Unless expressly stated otherwise in an official course document or handout, I expect that all work you submit to be your own. You may not share any source code files, design documents, or homework solutions. “Sharing” includes allowing (either actively or passively) someone access to your computer or to look at your screen where solutions might be displayed.

It is likely that you will do a significant amount of research throughout the course. Sometimes, it can be tempting to simply copy and paste code from a website without understanding it. The same is true if you choose to use any AI coding assistance tools like ChatGPT. But, I will caution you about mindless copying and pasting code from the internet into your projects.

1. Submitting other people’s intellectual work as your own is wrong.
2. Just because it is code on the internet does NOT mean it actually is described as doing. In other words, just because a block of code is described as a balanced binary search tree does not mean it is actually a balanced binary search tree.

If you find a small block of code online that you choose to incorporate into your project, make sure that you **fully** understand the code in depth. Read it carefully line by line and think about what’s happening.

You should be prepared to explain and defend in person any code or other deliverables you submit as part of an assignment for this class. Your submission is not considered accepted until Dr.

Fontenot determines an in-person interview is not needed.

I take academic integrity very seriously. **Therefore, the penalty for any act of cheating or academic dishonesty will be a failing grade in the course and submission of the matter to OSCCR.** I reserve the right to impose a less severe penalty at my sole discretion. Any penalties that OSCCR imposes will be separate from the course penalties.

Classroom Environment

Northeastern University values the diversity of our students, staff, and faculty, recognizing the important contribution each makes to our unique community.

Respect is demanded at all times throughout this course. In the classroom, it is expected that everyone is treated with dignity and respect. We realize everyone comes from a different background with different experiences and abilities. Our knowledge will always be used to better everyone in the class.

We strive to create a learning environment that is welcoming to students of all backgrounds. If you feel unwelcome for any reason, please let me or a TA know so we can work to make things better. If you feel uncomfortable talking to members of the teaching staff, please consider reaching out to your academic advisor.

Northeastern is committed to providing equal access and support to all qualified students through the provision of reasonable accommodations so that each student may fully participate in the learning experience. If you have a disability that requires accommodations, please contact the Disability Resource Center <http://www.northeastern.edu/drc/>, DRC@northeastern.edu, 617-353-2675. Accommodations cannot be made retroactively and to receive an accommodation, a letter from the DRC or LDP is required.

Schedule of Topics (Tentative):

Week:	Topics:
Week 1 (Sept 6-8)	Introductions & Syllabus Dev Environment Setup w/ Hello World
Week 2 (Sept 11-15)	Data Types and Operations Basic I/O Control Structures
Week 3 (Sept 18-22)	Built-in arrays and c-strings Writing Functions
Week 4 (Sept 25-29)	Parameter passing semantics Basics of memory
Week 5 (Oct 2 - 6)	Pointers and memory management
Week 6 (Oct 9 - 13)	Dynamic memory allocation and deallocation <i>Oct 9 - No class</i>
Week 7 (Oct 16-20)	OOP Basics in C++
Week 8 (Oct 23-27)	Operator Overloading Inheritance
Week 9 (Oct 30 - Nov 3)	Inheritance & Polymorphism
Week 10 (Nov 6 - 10)	Templates and Generic Programming
Week 11 (Nov 13 - 17)	STL
Week 12 (Nov 20 - 24)	Advanced Topics
Week 13 (Nov 27-Dec 1)	Final Project Work and Advanced Topics
Week 14 (Dec 4 - 6)	Final Project work and Advanced Topics

More information about the course project will be available later in the semester.