

Programming in C++ - 82496
CS 201 Section 02 Fall 2024
11:00 – 11:50 AM
Gries Hall 325 MWF

Professor: Jason Haskell

Office: Jamrich 2223

Email: jhaskell@nmu.edu

Regular Office Hours: Monday, Wednesday, Thursday 1:00-2:00pm

By Appointment Only Office Hours: Thursday 9:00am-12:00pm, Friday 9:00-10:30am

By appointment available in thirty-minute blocks on each hour and half hour. See ESP to make an appointment.

Text: No textbook will be used for this class. Students who desire a paper copy of reference material should purchase *C++ Pocket Reference: C++ Syntax and Fundamentals* by Kyle Loudon. Students will need to go to <https://visualstudio.microsoft.com/downloads/> and download Visual Studio 2022® Community for PC OR <https://code.visualstudio.com/download> and download Visual Studio Code® for Mac. Please be sure to install Visual Studio® if you are using PC the debugger is more robust for memory management.

COURSE DESCRIPTION

CS 201 teaches how to program in C++. Topics include pointers, memory management, data structures, object-oriented design and file processing.

Prerequisite: CS 122 or instructor permission.

Course Topics

1. C++ Syntax
2. Classes
3. Overloading Operators
4. Pointers and Memory Management
5. Recursive algorithms
6. File processing
7. Switch/Case
8. Try/Catch
9. Type Casting

Course Learning Outcomes

1. Demonstrate proficiency using C++ including:

- a. Creating and manipulating variables
 - b. Creating, managing, and using classes and their associated files
 - c. Using Switch/Case as a decision method for an algorithm
 - d. Using Try/Catch to manage exceptions
 - e. Overloading operators for classes
2. Demonstrate proficiency with pointer manipulation including:
 - a. Creating, allocating, modifying, and deleting pointers
 - b. Dereferencing pointers and passing pointers as arguments
 - c. Creating, allocating, modifying, and deleting 1-dimensional and 2-dimensional pointer arrays
 - d. Using recursive algorithms to modify and delete 1-dimensional and 2-dimensional pointer arrays
 3. Demonstrate file processing including:
 - a. Opening/closing a file
 - b. Reading to a file
 - c. Writing to a file
 4. Become proficient in recursive programming including:
 - a. Demonstrating the ability to create and use recursive methods as part of an algorithm
 - b. Creating recursive methods to manage pointer arrays
 - c. Demonstrate the ability to follow a recursive method and accurately predict its outcome with and without method parameters

Assessment Format: Specific information on each assessment is below.

- **Programming Assignments (30%):** Programming assignments include both works in progress and finished programs. Most work will be scaffolded or started in class. Some grades will be just for partial code.
- **Quizzes (10%):** This class will have a quiz typically every day. The quizzes will be done on EduCat. The quizzes will be used as a check to see how students are internalizing the concepts taught in class and to track attendance. They will typically consist of a single question. Some quizzes will be completion grades only while others will be graded with feedback.
- **Exams (40%, 4 total):** Each exam will consist of questions from the material discussed in class. A university-approved excuse is generally a prerequisite for rescheduling any test. All exams will be taken in class. Additional instructions will be provided prior to each exam. The last exam will be given during finals week on **Tuesday, December 10, at 10:00am** at our usual classroom location in Gries Hall room 325.
- **Final Project (20%):** The final project will be a program that allows you to demonstrate your proficiency with the concepts taught in this course. Expect to work on this project both in class and outside of class to meet the deadline.

Grading Scale (%): Students' course grade will be weighted according to the percentages outlined under Assessment Format. Corresponding grades based on a percentage are below.

100 – 90%A
89 – 80%B
79 – 70%C
69 – 60%D
59 – 0%F

Use of AI Writing Tools:

AI writing aids such as ChatGPT are categorically not permitted at any stage of coursework in this class. Use of these tools will be treated as academic dishonesty, constituting a clear violation of NMU's [Academic Dishonesty Policy](#).

Anyone caught cheating will receive a zero for the assignment and an F for the class.

How do I get help in the class?

1. See me during office hours or set up an appointment.
2. Visit the CS tutor in Jamrich 2311, see schedule on EduCat.
2. Meet up with a peer from the class.
3. Go to All Campus Tutoring (generally available on the weekends). Check their walk-in tutoring schedule at <https://www.nmu.edu/tutoring/>.

NMU's Non-Discrimination Statement:

Northern Michigan University does not unlawfully discriminate on the basis of race, color, religion, sex, national origin, age, height, weight, marital status, familial status, handicap/disability, sexual orientation, or veteran status in employment or the provision of services, and provides, upon request, reasonable accommodation including auxiliary aids and services necessary to afford individuals with disabilities an equal opportunity to participate in all programs and activities.

Anyone having civil rights inquiries may contact the Equal Opportunity Office, 502 Cohodas Hall, telephone number 906-227-2420.

ADA Statement:

If you have a need for disability-related accommodations or services, please inform the Coordinators of Disability Services in the Dean of Students Office at 2001 C. B. Hedgcock Building (906-227-1737 or disability@nmu.edu). Reasonable and effective accommodations and services will be provided to students if requests are made in a timely manner, with appropriate documentation, in accordance with federal, state, and University guidelines.

Subject to change with notice.