MTH 4300: Algorithms, Computers, and Programming II

Fall 2024

Course Number: 56474; Section: KTRA

Instructor: Jaime Canizales

Email: jaime.canizales@hunter.cuny.edu (will update to baruch email soon)

Github Link: `https://github.com/canizalesjaime/Mth4300_algorithms_and_programming`

Office Hours: By appointment.

Recommended Texts: S. Lippman, J. Lajoie, B. Moo, C++ Primer, 5th Edition, Addison-Wesley Professional, 2012 and B. Stroustrup, The C++ Programming Language, 4th Edition, Addison-Wesley Professional, 2013.

Meeting Time & Location: Tuesdays and Thursdays 2:55PM - 4:35PM, VC 6-125.

Prerequisite: MTH 3300 or CIS 2300, as well as at least one class in Calculus (MTH 2205, 2206, 2207, 2610, 2630, 3010). **Note: this class is not open to those who have credit for CIS 3100 or CIS 4100. If you have credit for CIS 3100 and have not already been in contact with me about this, please see me immediately.**

Software/Technology: A C++ compiler.

Learning Goals: Upon completion of this course students will be able to:

- make use of pointers;
- create classes (abstract data types);
- create constructors and destructors;
- write class methods;
- overload functions and operations (polymorphism);
- understand the notion and the implementation of inheritance;
- properly implement portions of the C++ Standard Template Library;
- and solve problems efficiently by constructing and implementing appropriate algorithms and data structures such as lists, stacks, and binary search trees.

This syllabus is likely to evolve as the term progresses.

Class Number	Date	Topics
1	8/29	Intro
2	9/3	Basics of C++
3	9/5	Basics of C++ and Functions
4	9/10	Control Flow and Operators in C++
5	9/12	Arrays and loops
6	9/17	Multidimensional Arrays
7	9/19	Recursion
8	9/24	Pointers
9	9/26	Dynamic Memory
10	10/01	Object Oriented Programming
11	10/03	No class
12	10/08	Midterm
13	10/10	Overloading Operators
14	10/17	Intermediate C++ Topics
15	10/22	More Object Oriented Stuff
16	10/24	Standard Template Library for Vectors
17	10/29	Linked Lists, Destructors, Arrow Operator
18	10/31	Linked List Part 2
19	11/05	Selection Sort and Big O Notation
20	11/07	File Stream, STL List and Iterators
21	11/12	Binary Trees
22	11/14	Midterm 2 review
23	11/19	Midterm 2(Classes covered(inclusive): 10/10-11/07)
24	11/21	Binary Search Trees
25	11/26	Binary Search Trees
26	12/03	Binary Search Trees
27	12/05	Stacks
28	12/10	Queues
29	12/12	Maps
30	12/19	Final(cumulative), time:330pm-530pm, room: b-14-235

Course Policy:

1. Attendance: You are expected to attend every class. Please arrive promptly at the beginning of class.

2. Classroom Demeanor: During class meetings, it is expected that you respect the class and your classmates. In particular, you should refrain from making insensitive remarks, talking at a disruptive volume, or spending time using phones or computers on non-class related activities.

3. Participation: The best way to learn programming is by doing, and you will spend a significant amount of class time each period reading or writing programs on your own. I will frequently pause class to ask you to try a problem or consider a question. When I do so, I expect you to try to discuss approaches, ask questions, or explain things to one another. I expect you to take these moments seriously. Thank you in advance for your cooperation

4. Assignments: Homework assignments will be assigned on Brightspace frequently. In addition, there will be some suggested exercises, not to be turned in.

5. Exams: There will be 2 midterms and a final (consult the class schedule for specific dates). According to department policy, any student who scores less than 50% on the final exam **will not receive a passing grade for the course.** The midterms and final exam will be given during in-person meetings.

6. Grading: Assignments = 15%
Midterms(2) = 52%
Final Exam = 33%

7. Grading Scale: A 93.00, A- 90.00, B+ 87.00, B 83.00, B- 80.00, C+ 77.00, C 73.00, C- 70.00, D+ 67.00, D 60.00, F ¡ 60. I don't intend to alter this scale, but I reserve the right to do so; if I do, it will almost certainly be in your favor.

8. Late Assignment Policy: For full credit, all assignments should be submitted by the due date, by 11:59 PM. I will take off 10 points from the assignment for each day late, but I will give you a total of three penalty-free days late throughout the term (for example, you could submit one assignment two days late, and another assignment one day late, without penalty; further late submissions would then be penalized at 10 points per day).

9. Accommodations: Baruch has a continuing commitment to providing reasonable accommodations for students with disabilities. Like so many things this term, the need for accommodations and the process for arranging them have been altered by COVID-19 and the safety protocols currently in place. Students with disabilities who may need some accommodation in order to fully participate in this class should contact Student Disability Services as soon as possible at disability.services@baruch.cuny.edu.

10. On Getting Help, and Artificial Intelligence: Feel free to come to office hours or email me at any time if you need assistance! For programming assignments (and quizzes and exams), copying of any code produced by ChatGPT, similar artificial intelligence, or other human or internet sources is prohibited. "Paraphrasing" of code, whereby code produced by artificial intelligence or another source is lightly changed, and then submitted as original, is also prohibited. Submitting such code without acknowledgment is an academic integrity violation. 2 Generally, I caution students to be very careful when getting help - certainly from Generative AI, but also from videos, from online sources, and even from friends and tutors. This is not a discouragement from using any of these resources, but it is crucial that you actively engage with them, rather than simply listen or read: take notes, ask questions, pause and try variations on your own, etc. As an experienced instructor, I have seen student after student fall into the following trap: - student gets help from a person or source that presents a solution to a problem, without much input from the student themself; - then, due to the clarity of the presentation, student feels like they understand the solution; - finally, student finds that they are unable to make any headway with closely-related problems, or even with reproducing the solution to the original problem. The middle step, the sensation of comprehension, can be very misleading. (And the last step typically happens during exams.) Generative AI is a neat tool for exploration, but it makes the trap exceptionally easy to fall into. While limited use

of AI is not prohibited for problem sets or studying, you are discouraged from using it - especially since the accuracy of its responses, while impressive, is far from perfect. There are ways in which AI can be used constructively for education, but if your use does not involve substantial quantities of code that you have written on your own, then you are at risk of getting zero or negative return from your "studying."

11. Collaboration: You are encouraged to think about how to proceed on programming assignments and projects with your colleagues. However, when you sit down to code them, it is expected that what you write will be entirely your own work. You are not to copy (or "closely paraphrase") your classmates work. Additionally, the following rule absolutely must be followed for coding assignments: You are to report sources and people that you consulted in writing your code (again, copying is NOT acceptable - this is meant for people and sources/websites which gave you hints or inspiration). Violation of this rule will result in a report of academic dishonesty being sent to the Office of the Dean of Students.

12. Academic Honesty: The Department of Mathematics fully supports Baruch College's policy on Academic Honesty which states, in part: "Academic dishonesty is unacceptable and will not be tolerated. Cheating, forgery, plagiarism and collusion in dishonest acts undermine the college's educational mission and the students personal and intellectual growth. Baruch students are expected to bear individual responsibility for their work, to learn the rules and definitions that underlie the practice of academic integrity, and to uphold its ideals. Ignorance of the rules is not an acceptable excuse for disobeying them. Any student who attempts to compromise or devalue the academic process will be sanctioned." Academic sanctions in this class will range from an F on the assignment to an F in this course. A report of suspected academic dishonesty will be sent to the Office of the Dean of Students. Additional information and definitions can be found at http://www.baruch.cuny.edu/academic/academic_honesty.html