

Instructor: Ari Mermelstein

Email address: mermelstein AT sci DOT brooklyn DOT cuny DOT edu

Course Web page: www.sci.brooklyn.cuny.edu/~mermelstein

Class meeting hours: ERQ6: Thursdays 6:30-10:00PM in room 130 F (WEB Building)

TY3: Tuesdays and Thursdays 3:40-5:20PM in room 130 NE (New Ingersoll)

Office hour and room: Tuesdays 5:30-6:30 in room 532 NE. I will also be available before and after class to answer questions. If these times do not work for you, please make an appointment, and I will try my best to find a mutually agreed upon time.

Required Textbook

Starting Out with C++, from Control Structures Through Objects, Eighth Edition by Tony Gaddis

ISBN: 0-13-376939-9

Prerequisite

CISC 1110 - Introduction to Programming Using C++.

A grade of C or higher in CISC 1110 is required to be able to take CISC 3110.

Information most needed from CISC 1110:

1. Conditional statements (including if-statements, switch statements, and the ternary operator)
2. Loops (including for-loops and while-loops)
3. Arrays (including how to declare arrays, how to iterate through arrays, and how to pass arrays to functions)
4. Functions (including void returning functions, functions that return a value, functions that take parameters by reference)
5. Parameter passing schemes (difference between pass-by-reference and pass-by-value)
6. How to work with files (both input and output)
7. How to use strings and the associated string library functions (e.g. find(), length(), +, +=, ==, etc.)
8. The idea of a simple class (Which we will relearn as “structs” soon enough).

Course Objectives

After successfully completing this course, students will be able to

1. Trace and write programs using object-oriented programming techniques.
2. Discuss, and program effectively with, the relative merits and consequences of compile-time and run-time memory allocation.
3. Use effectively the programming environment offered by a Unix-like system.
4. Implement recursive solutions to problems and demonstrate how recursion is implemented by tracing changes in the runtime stack.

Academic Integrity

The faculty and administration of Brooklyn College support an environment free from cheating and plagiarism. Each student is responsible for being aware of what constitutes cheating and plagiarism and for avoiding both. The complete text of the CUNY Academic Integrity Policy and the Brooklyn College procedure for policy implementation can be found at www.brooklyn.cuny.edu/bc/policies. If a faculty member suspects a violation of academic integrity and, upon investigation, confirms that violation, or if the student admits the violation, the faculty member *must* report the violation.

Non-Attendance Because Of Religious Observance

The state law regarding non-attendance because of religious beliefs is on p. 53 in the Bulletin. Please let me know now if you have to miss an exam (as far in advance as possible).

Center for Student Disability Services

In order to receive disability-related academic accommodations students must first be registered with the Center for Student Disability Services. Students who have a documented disability or suspect they may have a disability are invited to set up an appointment with the Director of the Center for Student Disability Services, Ms. Valerie Stewart-Lovell at (718) 951-5538. If you have already registered with the Center for Student Disability Services, please provide your professor with the course accommodation form and discuss your specific accommodation with him/her.

Important Dates For the Fall 2017 Semester

Friday, August 25 - First day of weekday class
Thursday, August 31 - Last day to add a class
Monday, September 4 - Labor Day. no classes
Thursday, September 14 - Last day to drop without a W grade
Tuesday, September 19 - Conversion day - Thursday schedule
Wednesday-Friday September 20-22 - No classes
Friday-Saturday September 29-30 - No classes
Monday, October 9 - Columbus Day, no classes.
Friday, November 10 - Last day to drop with a W grade
Tuesday, November 21 - Conversion day - Friday Schedule
Thursday-Sunday November 23-26 - No classes.
Wednesday, December 13 - Reading Day
Thursday- Wednesday December 14-20 - Final exams.

Other important dates:

Thursday October 5 and Thursday October 12, I will not be able to attend. The time will be made up, but the matter in which this will happen is yet undetermined.

Grades

First Test - 25%

Second Test - 25 %

Final Exam- 30%

Homework - 15%

Participation- 5%

Note: Participation really does mean participation. This is not a free 5 points. It must be earned.

Exam Dates - Tentative

The first exam will be held on Thursday October 19 .

The second exam will be held on Thursday November 16.

Each exam will last only 1 hour and 40 minutes, which is the time for each TY3 class meeting.

The final exam (not tentative) for TY3 will be on Thursday, Dec. 14 at **3:30 PM (NOT 3:40!)**

The final exam (not tentative) for ERQ6 will be on Thursday, Dec 14 at **6:00 PM (NOT 6:30!)**

Note: The final exam will be cumulative, and I am not allowed to give any more time than 2 hours.

Final grade calculation

Your letter grade will be determined as follows:

A+ : 98-100

A : 93-97

A- : 90-92

B+ : 87-89

B : 83-86

B- : 80-82

C+ : 77-79

C : 73-76

C- : 70 - 72

D+ : 67-69

D : 63 - 66

D- : 60 - 62

F : < 60

I do not curve final grades by many points. I may round grades up or give a few points based on merit, but I will not curve grades by tens of points.

Homework

Homework will be assigned every 1-2 weeks, and you will typically have 2 weeks to complete assignments. Assignments will typically include multiple files. **You must use the school's UNIX accounts to do your homework. No other operating system will be allowed.** (On the first week of class, I will show you how to work this operating system, and also how to connect to the school accounts from home. You don't necessarily have to do your homework at school). **Only code that is commented and documented, explaining what the code does will be marked fully correct. If I can't ascertain the correctness of your solution, I can not possibly grade it properly.**

A hard copy of your homework must be handed in in class on the day that it is due. The reason I am requesting this is so that I can write comments on your programs, indicating what is incorrect and/or suggestions for improvement. In the past, when I have accepted submissions by email, this was untenable, to say the least.

The homework assignments will be graded out of 10.

Late homework will not be accepted under any circumstances, unless you have spoken to me ahead of time, explaining why you need an extension. I am more than happy to grant extensions if you are honest with me.

In addition, there will be Codelab assignments periodically assigned. Your grade on all of the Codelab assignments combined will be averaged in as one homework. Information on how to register for Codelab is on the course website. If you took CISC 1110 in Brooklyn College, you need to just add a course, and registering a new account will not be necessary.

Topics List

1. Review of CISC 1110.
2. How to use UNIX and UNIX-like operating systems
 - How to create and delete directories (folders)
 - How to navigate in the file system
 - How to use *nano* or *vim* as an editor to write programs
 - How to link files and compile C++ programs in UNIX
 - How to display files.
 - How to list the files in a directory
 - How to redirect input and output.
 - How to use *ssh* and *sftp* to login to UNIX accounts remotely and to transfer files between computers.
3. Structures
 - How to define and declare structures
 - How to initialize structures
 - How to pass them to functions
 - How pass-by-value / reference works with structures
4. Pointers
 - How to declare pointers
 - What uses pointers have
 - How you can pass-by-reference using pointers instead of references
 - How to dynamically allocate memory using *new* and *delete*
 - What assignment of pointers means and what problems this may cause
5. C-Strings
 - What the C programming language's support for strings is (This is important to know for reasons of backward compatibility)
 - How to use the C library functions for C-strings (e.g. *strlen()*, *strcpy()*, *strcmp()*, *strcat()*, etc.)
6. Classes
 - What a class is
 - How to use access modifiers with member variables (*public* and *private*)
 - What accessor and mutator member functions are and what their purpose is
 - How to create a simple class with associated behavior
 - How to create constructors with and without arguments
 - How to use member initialization lists in constructors.
7. Operator overloading
 - How to overload boolean operators
 - How to overload mathematical operators

- How to overload the input and output (stream extraction and insertion) operators
- How to overload the assignment operator and copy constructor and why these are necessary

8. Exception handling

- How and when to use exception handling
- How the exception handling mechanism works (*throw* and *catch*)

9. Our own string class

- This will use what we learned about classes, C-Strings, and operator overloading to emulate the C++ string class.

10. Recursion

- What recursive functions are
- How to solve problems using recursive thinking

11. Inheritance and Polymorphism

- What inheritance is and how and when to use it
- What the access specifier *protected* is
- What polymorphism is and how to accomplish it in C++.
- What virtual functions are and how to create purely virtual functions