

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: MTWTH 8:00-9:45AM in room 431 NE (New Ingersoll)

Office hour and room: It will be easiest to just stay after class and talk to me. If this does not work, we can find a different time to meet in person or on Zoom/blackboard/etc.

Textbook

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

ISBN: 0-13-376939-9

To be honest, this is an introductory text, and a bit of an older one. This class will not be taught at introductory level; we all have programming experience in this class. So really, any C++ book will do, including free online textbooks, and I will not be teaching directly from the book.

Prerequisite

CISC 1115 - Introduction to Programming Using Java.

Information most needed from CISC 1115:

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. Methods
5. The Java memory model
6. How to work with files (both input and output)
7. How to use strings and the associated string library methods (e.g. indexOf(), length(), equals(), etc.

CISC 3115 – Modern Programming Techniques Using Java

Information most needed from CISC 3115:

1. The difference between primitive types and reference types.
2. What actually happens when you declare a reference to an object.
3. What “new” actually does.
4. Recursion

5. Classes- including:

- =accessors/mutators
- Constructors
- method overloading and overriding and the difference between those.
- In particular, overriding the *toString()* and *equals()* methods. -
- The difference between `==` and `.equals()` -
- Inheritences and polymorphism.

6. Interfaces

- How to define your own interface
- How to write a class that implements an interface (both your own and built in ones)

CISC 3130- Data Structures

Information most needed from CISC 3130:

1. Java Collections Hierarchy and how it is structured
2. ArrayList, LinkedList, TreeSet, TreeMap, HashSet, HashMap, PriorityQueue, stacks, queues. Their associated operations and running times.
3. Bonus: functional programming, lambda expressions, anonymous classes.

CISC 3310 – Principles of Computer architecture

Information most need from CISC 3310:

Introduction to assembly languages: ALU and memory reference instructions, flow control, subroutine linkage, arrays and structures. Memory. I/O systems. Performance. Relationship between software and architecture.

Course Objectives

After successfully completing this course, students will be able to :

1. Understand C++ and the use of and consequences of objects-as-values and direct access to memory management facilities.
2. Adept, efficient C++ programming.
3. Increased insight into programming paradigms in general and object-oriented programming in particular, as this second in depth study of an OOP language takes place.
4. Gain further insight into machine architecture and system organization through the use of this relatively low-level language.

5. Develop significant, efficient, flexible C++ application programs
6. Compare and contrast Object-Oriented programming in C++ and Java
7. Speak knowledgeably of the different programming paradigms.

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. [1] [SEP]

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 Summer 2022 Semester

Thursday, July 14 - First day of weekday class

Thursday, August 4- Midterm Exam

Thursday, August 18 – Final exam

Grades

Midterm- 35%

Final Exam- 35 %

Homework/Project- 20%

Participation- 10%

Note: Participation really does mean participation. This is not a free 10 points. It must be earned (especially since it's 10 whole points)

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.

Topics List

1. Writing the C++ equivalent of CISC 1115 code. This will include:
 - Loops
 - Conditionals
 - Arrays

- Functions (what C++ calls methods)
 - I/O
 - Parameter passing schemes and C++'s memory model
 - Comparing and contrasting with Java
2. 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
 3. 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
 4. 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.)
 5. 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.
 6. 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
 7. Exception handling
 - How and when to use exception handling
 - How the exception handling mechanism works (*throw* and *catch*)
 8. Our own string class
 - This will use what we learned about classes, C-Strings, and operator overloading to emulate the C++ string class.
 9. 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

10. The C++ STL

- How to use the built in data structures in C++.
- Solving interview type problems to illustrate this.
- How to use STL to write and make use of generic algorithms

11. Generic programming using templates.