CISC 3130 Data Structures Spring 2023

Instructor: Ari Mermelstein
Email address for questions: mermelstein AT sci DOT brooklyn DOT cuny DOT edu

Class meeting hours: Mondays and Wednesdays 9:05 – 10:45 AM in room 3413N (old Ingersoll)

Office hour and room: TBA

Required Textbook

Data Structures & Algorithms in JAVA by Goodrich, Tomassia, and Goldwasser 6th edition. ISBN: 978-1-118-77133-4.

or

Starting Out with Java, from Control Structures Through Data Structures, Third Edition by Tony Gaddis ISBN: 9780134038179

Essentially, if you already have the Gaddis book from last term, you don't have to buy another book. However, I find that Goodrich and Tomassia do a better job of talking about implementations of data structures than Gaddis, which is a harder thing to understand.

Prerequisite

CISC 3115 - Advanced Programming Techniques

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)

Course Objectives

After successfully completing this course, students will be able to

1. Demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, and vectors and be able to use these classes effectively in application programs.

2. Demonstrate ability to use the Java collections' data structures, and understand the advantages and disadvantages of each one, including the applications of the various data structures.

3. Discuss different possible implementations for various data structures in more than one way, compare the different implementations and explain the advantages and disadvantages of the different implementations.

4. Demonstrate understanding of and be able to program various sorting algorithms, and be able to compare the efficiency of these algorithms in terms of both time and space.

5. Trace and code recursive functions, especially those used to implement the data structures discussed above.

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.

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 2022 Semester

Tuesday, January 31: Last day to add a class.

Monday, February 13: College Closed.

Monday, February 20: College Closed.

Tuesday, February 21: Monday Schedule. We will meet.

Wednesday-Thursday April 5-13 – No class

Tuesday, May 16 – last day to W from the course

Grades

First Test - 25%
Second Test - 25 %
Final Exam- 30%
Homework - 15%
Participation- 5%

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

Exam Dates - Tentative
The first exam will be held on Wednesday, March 8.
The second exam will be held on Wednesday May 3
The final exam (not tentative) will be held on Monday May 22, 2023 from **8:00AM-10:00AM** (not 9:05!), room TBA (but usually in our regular room).

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 may use any operating system you like.
I would very much appreciate if you could print out your homework and submit them in class. This makes it much easier for me to grade your work and give feedback.

Please comment your code very well. If I have no idea what your code does, I can not possibly give you credit.

Topics List

1. Review of CISC 3115.
2. A rational number class and how generics work.
3. What the Java Collections hierarchy is.
4. The built-in ArrayList class and its methods.
5. The Iterator and Iterable interfaces; iterators for the ArrayList class.
6. Implementing our own MyArrayList class.
    -the difference between static and non static inner classes
7. Algorithm Analysis and Big Oh Notation.
    - how to analyze different algorithms and figure out how long they take to run
    - how to compare the running times of ArrayList and LinkedList operations.
8. The built-in LinkedList class and its methods.
    -why iterators for LinkedLists are essential.
9. Implementing our own MyLinkedList class - A ListNode class.
    - how to link these nodes into a linked list structure.
    - how to write methods to insert, remove, and print all nodes in a linked list. - doubly linked lists and the full MyLinkedList class.
10. A different approach for Lists -- Abstract classes. (Optional)
11. Stack and Queue interfaces
    - Specifying Stack and Queue interfaces and implementing our own classes that implement them.
    - Using the built-in Deque interface and its associated implementation classes as a way of writing programs that use stacks and queues in standard Java.
12. Using the built-in PriorityQueue class and its methods.
    - The difference between the Comparable and Comparator interfaces

- How to make our own custom Comparators (and a bunch of cool shortcuts).

13. Implementing our own priority queue.
    - surveying different implementations that we already know how to implement, and why these implementations aren't the best.
    -what a minimum heap is and how to implement it.
    - how to implement the methods insert() and removeMax().
14. Sorting algorithms and their complexities
    - what makes a sort inplace and stable.
    - which sorts are best in which situations.
15. The Java Set interface and HashSet and TreeSet classes.
    - what built in methods accomplish mathematical set-theoretic operations.
    - the difference in complexities for TreeSets and HashSets, and the difference in ordering.
    - an "interview type problem" that uses Sets.
16. The Java Map interface and the HashMap and TreeMap classes.
    - same complexity questions as above.
    - an "interview type problem" that uses maps.
17. Implementing maps and sets
    -Binary search trees
    -Hash Tables.
18. Graphs and Graph traversals.
    - Implementing graphs using the adjacency lists and adjacency matrix representations, and pros and cons of each.
    - Depth first search and breadth first search and their implementations.
19. If we have time:
    - functional programming
    - Using Java's interfaces and generics to create classes that behave like functions as first class objects.