CISC 3130 Data Structures Spring 2021

Instructor: Ari Mermelstein
Course Code: TY2
Email address for questions: mermelstein AT sci DOT brooklyn DOT cuny DOT edu
Email address for homework submissions:
mermelstein DOT homework AT gmail DOT com

Class meeting hours: Tuesdays 2:15-3:30 on Blackboard Collaborate

Thursdays 1:25 – 3:30 on Blackboard Collaborate

Office hour and room: Tuesdays 1:00 – 2:00PM on Blackboard Collaborate (or by appointment)

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 Spring 2021

Friday, January 29 - First day of weekday class
Thursday, February 4 - Last day to add a class.
Friday, February 12 and Monday, February 15 -- College Closed

Saturday, March 27 – Sunday April 4 – Spring Recess

Monday, May 17 – Last day to withdraw with a W.

Tuesday, May 18 – Reading Day

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 Thursday March 11.
The second exam will be held on Thursday April 29.

The exams will be held on blackboard.

The final exam (not tentative) will be on Thursday, May 20, 2020 at **1:00PM (NOT 1:25!)**
Note: The final exam will be cumulative, and I am not allowed to give any more time than 2 hours.

Exams will be given via blackboard during regular class time on the days listed above. You will not be permitted to backtrack and answer questions that you've already seen. Last term, many students googled answers, and I could tell. If this happens, I will give a 0 for that question.

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.

Homework must be submitted to Mermelstein.homework@gmail.com with the subject including CISC 3130 and your name. Or we may use blackboard.
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. Creating our own MyList interface.
7. Implementing our own MyArrayList class that implements the MyList interface.
8. Algorithm Analysis and Big Oh Notation.
9. The built-in LinkedList class and its methods.
   - why for LinkedLists iterators are essential.
10. 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.
    - An interview type question with linked lists
11. A different approach for Lists -- Abstract classes.
12. Stack and Queue interfaces

- Specifying Stack and Queue interfaces and implementing our own classes that implement them.
- Solving interview type problems with stacks and queues

13. Using the built-in Deque interface and its associated implementation classes.
14. Using the built-in PriorityQueue class and its methods.
15. Implementing a priority queue.
    -what a minimum heap is and how to implement it.
    a how to implement the methods insert() and removeMax().
16. Sorting algorithms and their complexities
17. An interesting interview type problem about sorting and priority queues.
18. A MySet interface and the HashSet and TreeSet classes.
19. A MyMap interface and the HashMap and TreeMap classes.
20. Implementing maps and sets
    - Binary search trees
    - Hash Tables.
21. Interview type problems with sets and maps
22. Graphs and Graph traversals