Course Syllabus

CS 2321 Data Structures

Spring 2023

Instructor: Ruihong Zhang

Email: ruihong@mtu.edu

Office Hours: MW 9:30-10:30AM TTH 11AM-12PM in Rekhi 302 Or by App

Teaching Assistant: See Course Home Page on Canvas

Schedule:	R01: Monday, Wednesday, Friday at 11am - 11:50 am in Fisher 139 R02: Monday, Wednesday, Friday at 1:00 - 1:50 in Fisher 138
Text: **Required**	*Data Structures and Algorithms in Java*, 6th edition by Goodrich and Tamassia, Willey. Inclusive Access Materials ($21) integrated in Canvas You may opt-out the Inclusive Access program and get your own physical copy in Campus Book Store For more information about Inclusive Access Program: https://www.bookstore.mtu.edu/michtech/site_textbook_faq.asp
Topics:	Proficiency in data structures (including containers, lists, heaps, priority queues, maps, hash tables, search tree, balanced search tree and graphs) and algorithms for manipulating data (including sorts, tree traversals and graph algorithms). Develop the ability to implement those data structures with analytic understanding of the time and space tradeoffs for different implementations. Solve problems efficiently with appropriate data structures.
Learning Objectives	Upon successful completion of this course, students will be able to: • Master the definition of the Abstract Data Structure of List, Priority Queue, Map, and Graph • Master different ways of implementing the ADTs including the data structures and the algorithms for implementing the methods • Master the following sorting algorithms: insertion sort, selection sort, merge sort, quick sort, heap sort and bucket sort. • Analyze the worst case and best-case time complexity for a given algorithm written in pseudocode with loops and recursive calls • Understand and Identify the programming technique of divide and conquer, greedy programming used in algorithms • Implement ADTs correctly in Java using different basic data structures and analyze the time complexity for ADT's operations • Conduct time testing for different algorithms/methods and analyze the results • Choose appropriate ADT to be used to solve other complex programming problems.
Prerequisite:	CS1122 or CS1131. Basic programming techniques and practices will not be taught. Know data structures at the level of arrays, linked lists, iterators, stacks and queues. Introduction to trees and binary search trees.
Homework:	There will be programming assignments and written homework assignments. **Programming Assignment:** Start early! Start early! Start early! Students may use any code in the assigned text or the lecture notes examples. Students may discuss the programs among yourselves without referencing your source code. The submission of program states that the work is your own.

<table>
<tr><td></td><td>Later assignments will depend on the abstract data type (ADT) implementations in the earlier assignments. Students are encouraged to use their own ADT implementations from the early assignments. When students have difficulties to correctly implement necessary ADTs, they could import the class that has been implemented by the instructors of the class. No source code solution will be provided for programming assignments.

<u>Written Homework Assignment:</u> The written homework is directly related to the concept and algorithm covered in lectures. The purpose of written homework is for student to practice the skills of choosing different data structures, describing the algorithm concisely with pseudo code and comparing the performance of different implementations.

<u>Due date:</u> Assignments are to be submitted before the due date. NO EXTENSIONS WILL BE GIVEN, except for an excused absence (https://www.mtu.edu/deanofstudents/academic-policies/attendance/) that prevents the student from submitting the assignments on time. Note that a planned event (such as hw, proj, exam due for other classes) occurring on the due date of the program does not prevent a student from completing and submitting early.

<u>Question about grading:</u> All questions and concerns about the homework grades are handled by the teaching assistant (TA) via email correspondence or office visit. Please notify the grader any grading questions within a week after the grades have been posted on Canvas.</td></tr>
<tr><td>Exams:</td><td>There will be 3 exams. The first two exams are evening exams, and the final exam is comprehensive and will be in the final week. The instructor is responsible for grading exams, so please contact with the instructor directly with any concerns within a week after the grade is posted.</td></tr>
<tr><td>Course Grading:</td><td>The course grade consists of two parts: homework assignments (including programming and written) and exams. The weight of each part:
1. Assignments (60%):
• About 7 Written homework assignments: 15%
• About 9 Programming assignments: 45%
2. Exams (40%):
• First exam: 10%
• Second exam: 10%
• Final exam: 20%
<table><tr><th>Grade</th><th>A</th><th>AB</th><th>B</th><th>BC</th><th>C</th><th>CD</th><th>D</th><th>F</th></tr><tr><td>Score</td><td>90-100*</td><td>85-89*</td><td>80-84*</td><td>75-79*</td><td>70-74*</td><td>65-69*</td><td>60-64*</td><td>0-59</td></tr></table>*: You must get at least 50% for both assignment and exam, and 60% for the total to pass the course.</td></tr>
<tr><td>Code Sharing</td><td>You may not:
• Use code written by someone else, except code form the current course text book or web site
• Discuss programming issues by writing or showing code to fellow students
• Use downloaded code from any web site (except this course’s current web site)
• Post your code on the web
You may:
• Discuss the programming assignment without reference to code
• Consult a learning center coach or instructor
• Discuss and ask about coding problems in class</td></tr>
<tr><td>CANVAS</td><td>Most course materials such as syllabus, slides, class notes, assignments, solutions, announcements, will be posted on Canvas.</td></tr>
<tr><td>Lecture Participation:</td><td>Attending class is essential for academic success. Students are expected to attend all scheduled class times unless an absence is excused under this university policy. Students are responsible for all material presented in class, which includes material that may not appear in Canvas or the textbook, and verbal changes in assignments.

In the case of excused absence, students are responsible to study and catch up with the class as soon as possible using the recorded lectures, slides, and textbook. If you have questions, please stop by during my office hour.</td></tr>
<tr><td>Disabilities:</td><td>Please get the accommodation approved by the Dean of Student Office.</td></tr>
</table>

Attendance Policy	Please read the Attendance Policy here: https://www.mtu.edu/deanofstudents/academic-policies/attendance/
Other University Policies	Please see Academic Integrity, Assessment, Disability Services and other policies in https://www.mtu.edu/ctl/instructional-resources/syllabus/policies/index.html
Other Resources	Java Tutorials: https://docs.oracle.com/javase/tutorial/ Eclipse: Java development user guide

Tentative Course Schedule for Spring 2023

W#	Days of the Week	No.	Topics	Homework Assigned	Prog Assigned
W1	M 1/9	1	Syllabus Fundamental Data Structures: Array (§3)	**HW1** Java Review	**Prog 0** Warmup
W1	W 1/11	2	Fundamental Data Structures: Singly Linked List (§3)		
W1	F 1/13	3	Testing, Junit and Debugger in Eclipse		
W2	M 1/16		Martin Luther King Day		
W2	W 1/18	4	Java Feature Review for CS2321(§1, §2)		
W2	F 1/20	5	Stacks and Queues (§6)		**Prog 1** Stack/Queue
W3	M 1/23	6	Prog1 Preview Application of Stack and Queue		
W3	W 1/25	7	Index Based List ADT (§7)		**Prog 2** List
W3	F 1/27	8	Doubly Linked List (§3) Position Based ADT (§7)		
W4	M 1/30	9	DLL Iterator, LRU, Prog2		
W4	W 2/1	10	Trees (§8)		**Prog3** Tree
W4	F 2/3	11	Trees (§8) (Cont.) Asymptotic Analysis (§4, §5)	**HW2** Analysis	
W5	M 2/6	12	Asymptotic Analysis (cont.)		
W5	W	13	Asymptotic Analysis (cont.)		

	2/8				
W5	F 2/10		Winter Carnival Recess		
W6	M 2/13	14	**Exam1 Review**		
W6	W 2/15	15	Priority Queues (§9.1 §9.2 §9.4) PQ List Implementation Heap (§9.3§9.4)	**HW3** PQ	**Prog 4** Heap PQ
	2/16		Evening Exam 1: Thursday 2/16 6-8PM in Fisher 135		
W6	F 2/17	16	Heap (Cont.) Adaptable PQ (§9.5)		
W7	M 2/20	17	Heap implementation pseudo code		
W7	W 2/22	18	Amortized Cost Analysis Prog3 Annotation and TCJ PQ sort Greedy Method (§13.4.2)	**HW4** Greedy	**Prog 5** Greedy Alg.
W7	F 2/24	19	Prog4 Discussion Huffman Coding Algorithm (§13.4)		
W8	M 2/27	20	Selection Sort Insertion Sort Merge Sort (§12)	**HW5** Sorting	**Prog 6** Sorting
W8	W 3/1	21	Quick Sort (§12)		
W8	F 3/3		No class due to evening Exam 1		
			Spring Break Recess		
W9	M 3/13	22	Quick Sort – cont. (§12) In Place Heap Sort §9.4.2		
W9	W 3/15	23	Sort Complexity (§11.3.1) Bucket Sort and Radix Sort (§11.3.2) Prog3		
W9	F 3/17	24	Maps (§10.1) Hash Table (§10.2)		
W10	M 3/20	25	Hash Table Cont. (§10.2)	**HW6** Map	**Prog 7** Map ADT
W10	W 3/22	26	Sorted Table		
W10	F	27	Binary Search Tree (§11.1)		

	3/24		**Exam2 Review**		
W11	M 3/27	28	Binary Search Tree (§11.1) (Cont.) Assignment 6 Discussion		
W11	W 3/29		No class due to evening Exam 2		
	3/30		Exam 2: Thursday 3/30 6-8PM in Fisher 135		
W11	F 3/31	29	AVL Tree (§11.3)		
W12	M 4/3	30	(2,4) Tree B Tree		
W12	W 4/5	31	Graph (§14.1 §14.2)		**Prog 8** Graph ADT
W12	F 4/7	32	Graph (§14.1 §14.2) (Cont.)		
W13	M 4/10	33	Assignment 8 Discussion Depth First Search (§14.3)		
W13	W 4/12	34	Breadth First Search (§14.3)	**HW7** Graph Alg.	**Prog 9** Graph Alg.
W13	F 4/14	35	Dijkstra's Algorithm (§13.5)		
W14	M 4/17	36	MST and More Graph Algorithms		
W14	W 4/19	37	Dynamic Programming		
W14	F 4/21	38	**Final Exam Review**		
			Final Exam		