CSE 274  
Data Abstraction and Data Structures

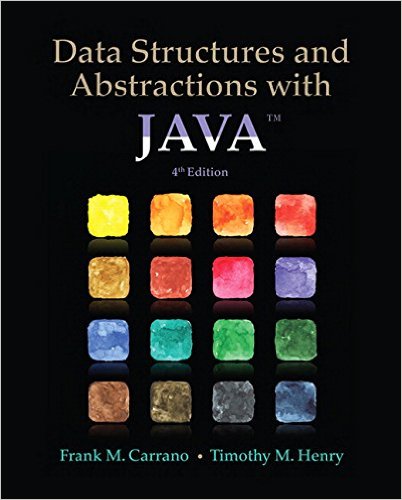
Summer 2017

**Instructor Information:**Name: Michael Zmuda (removed)  
Office: 201D Benton  
Office Hours: To be posted weekly on Canvas.

**TA Information:**Name: Chris Anderson (removed)

Office Hours: To be posted weekly on Canvas.   
  
**Course Details:**Days: MTR  
Time: 10:30am-12:35pm  
Location: 006 Benton  
Credit Hours: 3

**Catalog Description:** Abstract data types and their implementation as data structures using object-oriented programming. Use of object-oriented principles in the selection and analysis of various ADT implementations. Sequential and linked storage representations: lists, stacks, queues, and tables. Nonlinear data structures: trees and graphs. Recursion, sorting, searching, and algorithm complexity.

**Prerequisites:** CSE 271 with C- or above.  
  
**Coursepack:** Oxford Copy Shop has a set of printed notes that all students need. Alternatively, you can print the PDF available on Google Docs. Having a hard copy during class is needed when annotating source code, tracing algorithms, etc. Quiz #1 will require you to submit a specific page from the notebook.

**Textbook:** Frank Carrano and Timothy Henry. Data Structures and Abstractions with Java. ISBN-13: 978-0133744057. ISBN-10: 0133744051

Source code can be downloaded here: (removed)

Several appendices are available online. Links are provided in the outline.  
  
Prior students found this site useful for having visualizations: [Visualization website](https://www.cs.usfca.edu/~galles/visualization/Algorithms.html)

**Assignments:** The following table lists the weighting of the semester’s assignments.

|  |  |
| --- | --- |
| *Component* | **%** |
| ~6 Homeworks (equally weighted) | 40% |
| ~20 Laboratory assignments (drop lowest 4) Due at 1:00pm next calendar day. Late labs are not accepted. | 6% |
| 1 Quiz | 4% |
| Midterm and Final exam (equally weighted) | 50% |

Quizzes and Exams will be closed book, closed-note (unless otherwise stated). The final exam will be comprehensive and will take place during one of the class’ daily sessions. Missed exams/quizzes will be recorded as a zero unless other arrangements have been agreed upon. Live coding may be required.

Extra Time for Quizzes and Exams will be given to students who qualify for extra time, as determined by Miami’s SDS office. Students who qualify must take their exam with at the SDS office, starting at on the schedule day/time. Students are expected to schedule the exam at least 48 hours prior to the schedule exam. Exams will be taken at a time arranged during a meeting between student and professor.  
.

Laboratory Assignments. exercises require problem solving, short answer, design, analysis, and/or simple programming. The main differences between the lab exercises and regular homework assignments is that laboratory assignments are smaller, carry less weight, and are typically started and [possibly] completed during class. Lab assignments will be graded on a 10 point scale:

10 - correctly answers/solves the question/problem and is presented cleanly, clearly, and coded with elegance.

9 - correctly answers/solves the questions/problem, but has one stylistic/minor defect(s)

8 - correctly answers/solves the questions/problem, but has several stylistic/minor defect(s)

6,7 - has many stylistic defects and/or has at least one major defect

1,2,3,4,5 - has several major defects and/or does not fully address requirements

0 - work was not done

Homework assignments are written assignments completed outside of class requiring writing programs, problem solving, short answer, design, and analysis. These may resemble laboratory assignments but will be larger in scope.  
  
Attendance. Students are expected to attend class regularly. An attendance sheet will be provided in both sessions. Students are allowed 1 unexcused absences without penalty. Your course average decreases by 2% for each absence beyond these numbers. Being more than 5 minutes late to a class is considered ½ of an absence and being late more than 20 minutes is considered a whole absence. In accordance with university policy, I will drop you from the course if you have numerous absences. Please see me if you have a documented reason for your absence (e.g., doctor’s note).

**Final Grades:** Letter grades will be assigned based on the following scale. Numeric averages will be rounded to the nearest whole percentage.

|  |  |
| --- | --- |
| A+ | 97% ≤ x |
| A | 93% ≤ x < 97% |
| A- | 90% ≤ x < 93% |
| B+ | 87% ≤ x < 90% |
| B | 83% ≤ x < 87% |
| B- | 80% ≤ x < 83% |
| C+ | 77% ≤ x < 80% |
| C | 73% ≤ x < 77% |
| C- | 70% ≤ x < 73% |
| D+ | 67% ≤ x < 70% |
| D | 63% ≤ x < 67% |
| D- | 60% ≤ x < 63% |
| F | x < 60% |

**Policies:**

Office Hours: I maintain a sufficient number of office hours to assist students on an individual basis. Students are expected to utilize the office hours when difficulties arise. If you are unable to attend these office hours, we can schedule a meeting at a mutually agreeable time. Students are also encouraged to use email as a communication channel; especially long questions, however, should be addressed during office hours or class time.

The teaching assistant (TA): This course has a teaching assistant to perform grading and assist with office hours. If you have questions concerning grading, you are encouraged to contact the TA via email (CC the professor on these correspondences). If your concern is not addressed to your satisfaction, contact the professor directly.

Class preparation: Students are expected to keep current with the reading assignments. You should read the text, handouts, and supplemental materials carefully and bring questions to class on points that are unclear. Students are expected to spend ~15 hours outside class per week on the course material. This includes reading the textbook and solving homework problems. Many topics in this course are abstract and require this amount of work in order to fully comprehend the concepts.  
  
Canvas: I will make information and resources available on Miami’s Canvas system and the class’ Google Docs area. Students are required to look in these areas and download the notes on their own.

Late Work: Lab assignments cannot be submitted late. Late homework work will be penalized 5 points. Additionally, 2 points will be deducted for each hour late, where any fraction of an hour (i.e., [1-59] minutes) is treated as a whole hour late. Your submission can be at most 24 hours late.

Outside Help: Students are encouraged to discuss the course material with each other. On individual assignments, students must not explicitly exchange and/or share solutions to graded work (e.g., homework problems, programming assignments, exams, quizzes). Students are expected to use good judgment with regard to receiving outside help. If you receive considerable help from another student, write a short message acknowledging the help (e.g., “I am grateful to Mary Doe for suggesting the use of recursion”). Please refer to the Miami Student Handbook for a description of student expectations.

To help illustrate the expectations students will be held to, the CSE department has developed the following document:

[http://miamioh.edu/cec/\_files/documents/cse/general/AcademicIntegrity.pdf](http://miamioh.edu/cec/academics/departments/cse/academics/academic-integrity/index.html)

The problems assigned in this class are not research problems. That is, you are not allowed to search for solutions to the specific, or related, problems. Even if you plan to only view the solution, you are not allowed to do this. You may however use online resources to learn how to, for example, 1) declare a generic function 2) identify name and methods for Java’s priority queue 3) declare and throw your own exceptions 4) understand the specific attributes of an inner class. In all of these example, the online references do not address the crux of the assigned problem.

Also, be aware of the consequences of academic misconduct. The student handbook discusses the penalties associated with such conduct. <http://www.miamioh.edu/_files/documents/secretary/Student_Handbook.pdf>

Returned Work: Under normal circumstances, you will be allowed to retain your graded programs, homework assignments, quizzes, and exams. You will not, however, be allowed to retain your graded final exam.

Makeup Exams: Makeup exams/quizzes will be allowed under the following conditions: 1. An unexpected emergency arises - the makeup exam will be made up at the earliest possible time after the emergency. 2. The student has a university scheduled function which prevents attendance at the regularly scheduled exam - the makeup exam will be taken at the latest possible time that is prior to the regularly scheduled exam.

Miami University Learning Community: Miami University is committed to fostering a supportive learning environment for all students irrespective of individual differences in gender, race, national origin, religion, handicapping condition, sexual preference, or age. Students should expect, and help create, a supportive learning environment free from all forms of prejudice. Disparaging comments, sexist or racist humor, or questioning the academic commitment of students based upon these individual differences undermines our learning community. If such behaviors occur in class, please see the assistance of your instructor or department chair.

**Learning Outcomes**

1: Select and use appropriate data structures, abstract data types, and algorithmic methods in problem solving

1.1: Describe the purpose and semantics of abstract data types, including: matrices, lists (aka sequences), stacks, queues, sets, maps (aka dictionaries), trees, graphs, and priority queues

1.2: Describe the purpose and semantics of major data structures including: row-major order representation of matrices, 2D-array representation of matrices, array-based lists, linked lists, hash tables (both chaining and open addressing), binary search trees, adjacency lists, adjacency matrices, and heaps

1.3: Create programs that utilize the major data structures

1.4: Use the appropriate data structures for solving search problems

1.5: Describe the purpose and possible alternative implementations of common tree and graph algorithms including tree traversals (inorder, preorder, and postorder), depth first search, breadth first search, Dijkstra’s algorithm, topological sort, minimum spanning trees, and all-pairs shortest paths.

2: Implement common data structures and algorithms

2.1: Implement common data structures, including: row-major order representation of matrices, 2D-array representation of matrices, array-based lists, linked lists, hash tables (both chaining and open addressing), binary search trees, adjacency lists, adjacency matrices, and heaps

2.2: Implement common tree and graph algorithms including tree traversals (inorder, preorder, and postorder), depth first search, breadth first search, Dijkstra’s algorithm, topological sort, minimum spanning trees, and all-pairs shortest paths.

2.3: Combine multiple data structures to create an efficient solution to a problem (e.g. implementing an LRU cache using a list plus hash table, or efficiently changing the priority of an item in a heap in Dijkstra’s algorithm using an auxiliary array)

2.4: Create or modify one’s own classes to be used with library collection classes (e.g. create a class that can be used as a key in a hash table or binary search tree)

3: Implement data structures and abstract data types using object-oriented programming principles

3.1: Create data structures using polymorphism

3.2: Create data structures using inheritance

3.3: Create data structures using generic programming

3.4 Represent abstract data types using interfaces or abstract classes

4: Determine time and space requirements of common data structure implementations and algorithms

4.1: Describe the purpose of asymptotic notations (e.g., O) for algorithm analysis

4.2: Apply asymptotic notations (e.g., O) to analyze algorithms

4.3: Describe the differences between worst-case running time, expected running time, and amortized running time and apply them appropriately

**Tentative Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **M** | **T** | **R** |
| **1 6/26** | **†**Introduction Organizing Data  **†**Prelude Designing Classes  **†**Appendix Documentation and Programming Style  **†**Appendix B Java Basics (online)  **†**Appendix C Java Classes (online)  **†**Appendix D Creating Classes from Other Classes  **†**Appendix E File Input and Output (online)  JUnit  Chpt 1 Bags | Point/Triangle/Polygon  JUnit | JI 1 Generics  **†**JI 2 Exceptions  Chpt 2 Bag Impls. That Use Arrays  Chpt 3 A Bag Implementation That Links Data |
| **2 7/3** | **Quiz** Chpt 4 The Efficiency of Algorithms | **No classes. 4th of July Holiday** | Chpt 5 Stacks  Chpt 6 Stack Impls |
| **3 7/10** | JI 3 More About Generics  **†**Chpt 7 Recursion  **†**Chpt 18 Searching **†**Chpt 8 Introduction to Sorting  **†**Chpt 9 Faster Sorting Methods  **†**JI 4 More About Exceptions | Chpt 10 Queues, Deques, and Priority Queues  Chpt 11 Queue, Deque, and Priority Queue Impls. | **Midterm exam** |
| **4**  **7/17** | Chpt 12 Lists  Chpts 13, 14 List Impls  JI 5 Iterators Chtp 15 List Iterators  JI 6 Mutable/Immutable objects  Chpt 16 Sorted lists | J7 Inheritance and Polymorphism  Chpt 17 Inheritance and Lists  JI 8 Generics Again | Chpt 19 Dictionaries  Chpt 20 Dictionary Impls. |
| **5**  **7/24** | Chpt 21 Introducing Hashing  Chpt 22 Hashing as a Dictionary Impls. | Chpt 23 Trees  Chpt 24 Tree Impls. | Chpt 25 Binary Search Tree Impls.  Chpt 26 Heap Impls. |
| **6**  **7/31** | Chpt 27 Balanced Search Trees  Chpt 28 Graphs | Chpt 28 Graphs  Chpt 29 Graph Impls. | **Final Exam** |

**† are readings that are predominantly review  
Chpt == Chapter  
JI == Java Interlude**

**Impls == Implementations**