Lamar University
College of Arts and Sciences
Computer Science Department

COSC 2336-01

Programming Fundamentals III - Data Structures
Spring 2009

CONTACT INFORMATION: **1/6/2009**

Name: Myers L. Foreman
Phone: 409-880-8553
E-mail: Use the Blackboard Mail tool for communication. Programs and written assignments are to be sent to my grader using the Blackboard submission process. Failing that, they can be emailed to my grader if necessary. I will give you the name of my grader as soon as I know it.
Alternate E-mail: foremanml@my.lamar.edu (Only to be used if the Blackboard mail server is down.)
Office Location: O 75 Maes Building (Oddly enough, it is in a corner on the 2nd floor.)
Office Hours: 8:00 – 9:00 MWF
10:00 – 11:00 MWF

Monday	Tu	Wednesday	Th	Friday
Jan 12 First Class Day	13	**Jan 14**	15	**Jan 16**
Jan 19 No Class	20	**Jan 21**	22	**Jan 23**
Jan 26	27	**Jan 28**	29	**Jan 30**
Feb 2 Prog 1 Due	3	**Feb 4**	5	**Feb 6**
Feb 9	10	**Feb 11**	12	**Feb 13**
Feb 16 *First Drop*	17	**Feb 18**	19	**Feb 20**
Feb 23 Prog 2 Due	24	**Feb 25**	26	**Feb 27**
Mar 2	3	**Mar 4**	5	**Mar 6**
Mar 9 No Class	*10*	***Mar 11 No Class***	*12*	***Mar 13 No Class***
Mar 16 Prog 3 Due	17	**Mar 18 Mid-Term Exam**	19	**Mar 20**
Mar 23	24	**Mar 25**	26	**Mar 27**
Mar 30 *Last Drop*	31	**Apr 1**	2	**Apr 3**
Apr 6 Prog 4 Due	7	**Apr 8**	9	***Apr 10 No Class***
Apr 13	14	**Apr 15**	16	**Apr 17**
Apr 20	21	**Apr 22**	23	**Apr 24**
Apr 27	28	**Apr 29**	30	**May 1**
May 4 Prog 5 Due Last Class Day	5	**May 6**	7	**May 8**
May 11	12	**May 13 Final Exam. 8:00 – 10:30**		

CREDIT HOUR ALLOTMENT: This is a 3 credit hour course required of all CS and CIS majors.

PRE-REQUISITES & CO-REQUISITES: Completion of COSC 1337 or COSC 3306 with a grade of B or better. Successful completion of Math 2413 and Math 2305.

CATALOG COURSE DESCRIPTION: Further applications of programming techniques, introducing the fundamental concepts of data structures and algorithms. Topics include recursion, fundamental data structures (including stacks, queues, linked lists, hash tables, trees, and graphs.), and algorithm analysis.

STUDENT LEARNING OUTCOMES: After successful completion this course, the student will be able to:

1. Create useful software architecture documentation at the program, class, method, and block levels.
2. Develop correct and efficient programs to implement software.
3. Debug implemented software in a proficient manner.
4. Demonstrate familiarity with standard searching and sorting algorithms and linear and non-linear structures.
5. Define and distinguish among frequently used discrete structures such as lists, trees, and graphs to computer science problems.
6. Use elementary concepts of combinatorics, probability, and statistics to analyze and evaluate the efficiency of algorithms.
7. Demonstrate basic understanding of asymptotic notations and time complexity.
8. Design efficient algorithms and compare competing designs.
9. Demonstrate basic understanding of some design approaches such as greedy algorithms, dynamic programming and divide-and-conquer.

REQUIRED TEXT: No book is required, but this book is recommended as a reference. Many others are perfectly suitable alternatives however. Most of the teaching materials used in the class will be stored on the blackboard website.

(recommended Object Oriented Data Structures using Java (2nd ed), Dale, Joyce & Weems

TEACHING STRATEGIES:

- This is not an online course, but all the course materials will be placed online at the blackboard website. Furthermore, assignment submissions and assignment grade reporting will take place online.
- Lectures on specific Java language features such as arrays, string processing, primitive versus reference types, File I/O, inheritance, exception handling.
- Lectures will be provided on various construction strategies for stacks, queues, linked lists, trees, and graphs.
- In additions. various sorting and searching algorithms, basic algorithm analysis, and a number of basic algorithms styles will be introduced.
- A large number of Java example programs demonstrating various topics will be provided.
- 5 programming assignments will be made.
- Two exams will be given.

EVALUATION:

- Programming Assignments: 1/3 (Due dates are given in the calendar.)
- Test 1: 1/3 Wednesday, March 18
- Final: 1/3 Wednesday, May 13 (8:00 – 10:30)
- Program due dates, drop dates, and class holidays are listed in the calendar.

GRADING SCALE: No curve is guaranteed. If I feel a curve is justified at the end of the semester, then one will be applied to your final average. The curve will NEVER be larger than 10 points and is typically a point or two.

- A (90-100)
- B (80-89)
- C (70-79)
- D (60-69)
- F (0-59)

GRADING: I will grade the tests and my grader will grade the program assignments. Any concerns you have about my grader's efforts should first be directed to him or her. If you cannot get a satisfactory response there, then contact me. Tests should be graded within one week, and programming assignments within two weeks. There is a **GRADING** folder on the main page in blackboard which contains: **Guidelines for Grading Programming Assignments**. This is the same file that will be given to my grader, so I suggest you read it.

EXTRA CREDIT: Many programming assignments have an extra credit component that you may decide to complete. The amount of extra credit per assignment is usually in the 10 to 20% range. Extra credit will not be given for a program that would otherwise have received a grade less than 70%.

PROGRAMMING ASSIGNMENTS:

- Programs will be written in Java, and must run on a Linux or Unix system. My grader will be using one of the Linux systems in the CS department, so it is best that you test your programs there. You will already have or be given a userid and password that will work on any of our machines.
- Assignments will be submitted using the Blackboard submission process.
- Each assignment is due before midnight on the day specified.
- There are 5 programming assignments. Due dates are approximately every three weeks.
- The calendar above lists the actual due dates.
- No late assignments will be accepted except in special circumstances. Assignment instructions are located in Blackboard under the Assignments tab. Surprise.
- At the risk of belaboring the obvious, it is foolish to turn in nothing at all.

WHAT TO TURN IN:

Five (5) program assignments will be made. The files below will be attachments to your blackboard submission. Do not wait until the last minute.

1. Your submission should include your name, student id, and program number.
2. Attach all your source files.
3. Attach any data files.
4. My grader will compile and run your program under LINUX. Make sure your program will work there.
5. In addition to 2 and 3 above, attach script file containing:
    - All your source code

- Any data files
- Compilation of each source file
- Execution of the main
- If your program outputs to a file, be sure the script includes that also

The script fill will be useful if it appears your program will not run for my grader when you say it will run for you.

6. Last semester was the first time I tried to do a programming class without paper, but it seemed to go fairly well. Please let me or the grader know if you are having problems. I will give you the grader's name as soon as I know it.

EMAIL AND DISCUSSIONS

- Adhere to the same standards of behavior online that you follow in real life when writing emails or posting on the discussion board
- Use the discussion board to ask class or assignment questions
- I will respond to emails and discussion postings at least once a day during the week.

COMPUTER REQUIREMENTS:

If you do not have a computer there are many within the Maes building and the Lamar library that you can use. If you have a computer but no internet access, I can make you a copy of all class materials if you will bring me a flash drive or blank CD. In such a case, it will be necessary for you to submit your work using machines here.

Students who take courses via Blackboard will need to have the following capabilities and Software:

- High Speed (Broadband) Internet access is desirable
- Computer with at least 250 mb memory
- Word processor software (Compatible with Microsoft Word)(http://www.openoffice.org)
- Adobe Acrobat Reader (http://www.adobe.com)

* Center for Distance Education will help give you advice about what you need to consider for good quality computer hardware and software. Please call them at 880-7849.

View the Blackboard overview, located on the homepage, if you are new to online courses.

ON-LINE COURSE SUPPORT:

- Blackboard can be accessed directly at http://luonline.blackboard.com
- The course materials are being offered on-line via Blackboard to facilitate student access.
- The Distance Education Office will provide technical support for the course.
- There is 24 hour access for help through voice mail at the following Center for Distance Education number **(409) 880-7849**.

Please use these resources to assist you with any technical problems that may develop.

LIBRARY SERVICES:

Library services for online students can be accessed at http://library.lamar.edu/

READINESS FOR ON-LINE COURSES:

Students may access Blackboard at

1. http://luonline.blackboard.com
2. http://my.lamar.edu through the STUDENTS tab
3. From the Lamar website at http://www.lamar.edu

Regardless of how you enter:

- Your *username* is the same as your myLamar username (see http://my.lamar.edu for details)
- Your *password* is your student id number.

COSC 2336 Topic Outline   Data Structures

I. Linear Structures

1. Stack
2. Queue and Deque
3. SLL - Single Linked List with variations
4. DLL - Double Linked List with variations

II. Implementation Schemes for Data Structures

1. Language Primitive/Defined (Show variations among languages.)
   a. Integer
   b. Real
   c. String (Static, Dynamic, and Hybrid)
   d. Set
   e. Record, Struct, Heterogeneous Aggregate
   f. Array, Homogeneous Aggregate
   g. Others, (Stack, List, Logical, Complex, etc.)
2. Programmer Defined
   a. Static with Implicit Pointers (Typically using an array.)
   b. Dynamic with Explicit Pointers (Typically using dynamic variables.)
   c. Static with Explicit Pointers (Typically using static instance data.)
3. Language Assisted

   Through interfaces and imports, modern languages such as Java provide support for lists, array lists and trees containing any sort of object, as well as sets, hash tables, etc.

III. Non-Linear Structures

1. MLL - Multi-Linked List
2. Binary Tree
   a. Pre-Order, Post-Order, In-Order, and Level-Order Traversals
   b. Binary Search Tree (BST), insertion and removal
   c. Expression Tree
   d. Heap
3. M-ary Tree
4. Huffman Algorithm for Text Compaction
5. M-ary Tree implemented as a binary tree
6. AVL Tree (Height Balanced BST) insertion only
7. B tree, B+ Tree
8. Game Trees (Alpha-Beta Pruning if time permits)
9. Optimal Binary Search Tree
10. Graphs
    a. Directed
    b. Undirected

- c. Graph Implementation Schemes
    - i. Connectivity Matrix
    - ii. Weighted Connectivity Matrix
    - iii. N union E as an MLL
    - iv. Array-Array and Array-List combinations
    - v. Efficient storage for sparse matrices

11. Graph Algorithms
    - a. Breadth First Traversal
    - b. Depth First Traversal
    - c. Kruskal's Minimum Spanning Tree of a Graph
    - d. Dijkstra's Single Source Shortest Paths Algorithm
12. Heap implementations (with heap sort)

IV. Internal Sorting

1. Bubble, Selection, Insertion, etc.
2. BST Sort
3. Heap Sort (max-heaps and min-heaps)
4. Quick Sort
5. Bin or Radix Sort using LLs
6. Address Calculation Sorting Techniques
7. Merge Sort using LLs

V. Searching

1. Linear Search
2. Binary Search
3. Binary Search Tree
4. Hashing and Symbol Tables
5. Open or Bucket Hashing
6. Closed Hashing (Linear Quotient)

VI. Algorithmic Analysis, Styles and Techniques

1. Recursion (Direct and Mutual)
2. Worst Case Analysis (Big O)
3. Divide and Conquer
4. Greedy Method
5. Exhaustive Search
6. Backtracking
7. Dynamic Programming
8. Introduction to Stochastic and Event Driven Simulation

VII. Notes

1. COSC 3304 is titled Algorithm Analysis and Design. It is designed to follow 2336. Although some of the material will overlap, 3304 will discuss algorithm analysis and efficiency in much greater and more rigorous detail than 2336.
2. The grouping above is logical and does not coincide with the order in which the material is taught. Material order is dictated in large part by programming assignment requirements.
3. The concept of a heap is saved until the heap sort is discussed.
4. The BST sort is used as a tree example before the bulk of the other sorts are discussed.
5. The algorithm analysis material is discussed as needed, primarily during the discussion of sorting and searching techniques.
6. As recursion techniques are particularly useful for tree algorithms, recursion is primarily discussed in conjunction with this topic. Some of the possible applications of recursion to linear structures are also discussed.

7. Program assignments are designed so that the student must implement data structures using all three basic implementation schemes.

ACADEMIC HONESTY:

The university expects all students to engage in all academic pursuits in a manner that is above reproach.

“Cheating” includes:

1. Copying from another student’s paper, report, computer files, data listings, and/or programs.
2. Using (during an exam), materials not authorized by the faculty giving the exam.
3. Collaborating, without authorization, with another person during an examination or in preparing academic work.
4. Knowingly, and without authorization, using, buying, selling, stealing, transporting, soliciting, copying, or possessing in whole or part, the contents of an un-administered test. Substituting for another student; permitting any other person; or otherwise assisting any other person to substitute for oneself or for another student in the taking of an examination or the preparation of academic work to be submitted for academic credit.
5. Bribing another person to obtain an un-administered exam or information about an un-administered exam.
6. Purchasing, or otherwise acquiring and submitting as one’s own work any research paper or other writing assignment prepared by an individual or firm. This section does not apply to the typing of the rough and/or final versions of an assignment by a professional typist.
7. Any copying from library or other resources, including the Internet/WWW, without the instructor’s prior knowledge and approval, or without giving (clearly and conspicuously) the proper credit reference.

Plagiarism: The appropriation of another’s work or idea and the unacknowledged incorporation of that work or idea into one’s own work offered for credit.

Collusion: The unauthorized collaboration with another person in preparing work offered for credit.

COURSE ATTENDANCE POLICY:

A seating chart will be made and checked at the beginning of each class period. A student who is tardy or absent without excuse more than 4 times will be penalized three(3) points for each additional absence or tardy. If at any point, you decide to drop the class, it is your responsibility to officially drop. The drop dates enforced by the university are on the calendar. Any student who stops attending class and does not officially drop the course will be given an “F” as the semester grade.

UNIVERSITY DROP/WITHDRAWAL POLICY:

March 30 is the last day to drop the class. I am not sure how much trouble it is to drop from out of town, so don’t wait until the last minute if you are considering dropping. If you are out of town, and send me an email indicating your desire to drop the class early enough, I probably can arrange it.

STUDENTS WITH DISABILITIES:

If you have a need for a disability-related accommodation, please contact the Office of Services for Students with Disabilities (SFSWD), Room 101A, in the Wimberly Building on the Lamar campus.

SPECIAL ACCOMMODATIONS:

It is the policy of Lamar University to accommodate students with disabilities pursuant to federal and state law, and the University’s commitment to equal educational opportunities. Any student with a disability, who needs accommodation, should inform the instructor at the beginning of the course.