

# **CSC 443: *Data Structures***

## **Syllabus**

### **Winter 2010**

#### **Table of Contents**

- [Number and Title of Course](#)
- [Catalog Description of Course](#)
- [Instructor](#)
- [Required Textbook](#)
- [General Objectives](#)
- [Specific Objectives](#)
- [Major Topics](#)
- [Instructional Methods and Techniques](#)
- [Assignments for Course](#)
- [Evaluation](#)
- [Course Outline](#)
- [Ground Rules for Programming Assignments](#)

#### **Number and Title of Course:**

CSC 443 Data Structures

#### **Catalog Description of Course:**

Linear lists, stacks, queues, sequential and linked allocation of storage, circular lists, applications. Binary and ordered trees, traversing and threading trees, garbage collection, multilinked structures, dynamic storage allocation, data packing, hash coding. Computer projects.

## Course Prerequisite:

Students should have an extensive background in programming, including an extensive knowledge of and object oriented programming in C++ as taught in *CSC 441 Object- Oriented Programming* (formerly CSC 345) or CSC 172 *Computer Science II*

## Instructor: Dr. R. Michael Canjar

**Office Hours:** will be held in Engineering 323, Tentative schedule:  
Monday: 11:00 AM – 1:30 PM  
Tuesday: 11:00-11:30 AM,  
Wednesday: 11:30 AM – 1:30 PM

If I am not in, please check the computer lab across the hall.(E372)

**Phone:** (313)-993-1209

**Internet Usage:** We will use the Blackboard software which is available at <http://knowledge.udmercy.edu/>. Blackboard will be used communication (e-mails and announcements) and for distributing class handouts and examples, and grades

Please check that that your current e-mail address is registered in Blackboard. You may send me an e-mail from Blackboard; however you should make a note of the following address , in case Blackboard is down.

**E-Mail:** [CANJARRM@udmercy.edu](mailto:CANJARRM@udmercy.edu)

E-mail is the most efficient way to reach me. Please put a note in the subject line to indicate that it concerns the CSC 443.

Please send me an e-mail from the e-mail account that you plan to use. I will send you a confirmation. Let me know if you do not receive a confirmation from me, it probably means that spam-blocker stopped. Let me know an I will “white-list” you e-mail address. You should not need to do this if you use your students.udmercy.edu account.

## Required Text:

Mark Weiss *Data Structures and Algorithm Analysis using C++*, 3rd Edition.  
Addison-Wesley, ISBN: 0-321-37531-9  
<http://www.aw-bc.com/catalog/academic/product/0,1144,0321375319,00.html>

There will also be supplementary lecture notes and handouts.  
Texts used in pre-requisite courses may be valuable references.

## Software:

The in-class examples and projects were originally developed for Microsoft Visual C++. My understanding is that copies of a recent version of Visual C++.Net are available on to students on campus. The projects themselves were originally developed for version 6.0 of MS Visual C++.

I will endeavor to support both versions. I will give you a header file *443-rmc1.h* which will detect which version you are using and include the proper header files. If you are using version 6.0, it will include the pre-standard files. If using the more recent software, it will use the *namespace std* and include the corresponding headers.

## General Objectives

1. To understand the basic data structures: linked lists, stacks, queues trees, hash tables, and to understand their relative advantages and disadvantages for various algorithms.
2. To understand the key concepts of algorithm analysis, complexity theory, and the relationship between data structures and the various algorithms they support.
3. To provide more experience in programming and problem solving, and in particular on the use of object-oriented methodologies.

## **Specific Objectives:**

Upon completion of the course, students will

1. Be able to implement the elementary data structures.
2. Be able to use these data structures in specific applications, and be able to choose data structures which are appropriate for those applications.
3. Be able to analyze algorithms and to select the best algorithm for a specific problem.
4. Be able to use recursion to implement specific algorithms.
5. Be able to use existing classes and existing class libraries to implement required data structures.

## **Major Topics:**

1. Implementation and Applications of Linked Lists
2. Introduction to Queues and Stacks, and the Applications to Parsing.
3. Complexity Theory and the Analysis of Simple Algorithms.
4. Recursive Algorithms.
5. Algorithms for Sorting and Searching.
6. Trees, Basic Algorithms and Implementation.

## **Instructional Methods and Techniques:**

1. There will be lectures of 75 minutes twice a week.
2. Software and Example programs will be demonstrated in class. Examples will be made available on class handouts and may be downloaded from the Internet.
3. Students will have 3 or 4 lengthy programming assignments to be done outside of class. Extended Office Hours will be held in the Lab when necessary to provide additional help in 3.

## **Assignments for the Course:**

The traditional syllabus for this course specified:

1. There will be readings from the Text supplemented by class handouts.
2. There will be approximately 3 major programming assignments, some of which may be in several parts.
3. There will be an in-class Midterm and Final Exam. The current plan is for this exam to be open-book open notes, closed computer, and closed friends.

## **Course Evaluation**

The traditional syllabus for this course specified:

There be 2 Exams, a midterm and a final. The Midterm will count for 20% of the grade; the final will count for 40%. The remaining 40% will be based upon the programming assignments.

The current plan is for this exam to be open-book open notes, closed computer, and closed friends.

## **Attendance/Participation :**

Students are expected to attend class on a regular basis and participate in the discussions. They are responsible for all the material presented therein. Formal attendance records will not be maintained; however attendance is highly correlated with performance on the projects and the exams.

The instructor will attempt to make reasonable accommodations for students who miss a class due to illness, death in the family, or other legitimate reasons. However students who are forced to miss several classes will have difficulty completing the course in a satisfactory manner.

## **Academic Integrity**

Students are expected to conform to a high standard of honesty and integrity in this course. Copying the work of someone else and other forms of cheating are strictly prohibited. Permitting or tolerating such behavior is also prohibited. The minimum penalty for any offense is a 0 on that assignment. The culprits may be subject to additional sanctions, up to and including expulsion from school for serious offenses, as prescribed by the University Catalog and the Engineering Science Student Handbook.

For the group programming assignments, you obviously are free to cooperate with members of your own team. But the sharing of code between teams is explicitly forbidden.

## **Tentative Course Outline**

1. In Project 1, we will develop a polynomial class, using Linked Lists. This project will be due in approximately 4 weeks, on Thursday February 11
2. In project 2, we will build a polynomial calculator, which will include a Parser, using the Polynomial class, stacks, and a hash table. Part A of this will be due on Saturday Feb 26. Part B will be due on Thursday March 11.
3. There will be a third assignment which will involve trees and recursion, due during the last week of class, approximately April 22.
4. The Final Exam is scheduled for Monday April 26 at 2 PM

## **Make Up Policy :**

Make Up exams will only be given to students who miss an exam for legitimate reason (as defined above under "Attendance") and who notify the instructor in advance.

## Other Important Dates

You can also read the tentative academic calendar online. Look at AY 2009-2010 which now is the first column <http://www.udmercy.edu/registrar/academic-calendar/1012calendar3yr.pdf>.

Some important dates to note are

January 15	Last Day to Add or Delete a Class
January 18	Martin Luther King, Jr. Holiday (No classes/Offices Closed)
March 2	Mid-term grades Due from Faculty
March 8 - 13	Mid-Winter/Spring Break (No classes/Offices Open)
April 1	Last Day to Withdraw from Class for Winter
March 28	Honors Convocation
April 2-4	Easter Recess - University Closed
April 26-May 1	Final Exam Week
May 1	Official End of Term II Winter 2008-2009
May 3	Grades Due for Term II Winter 2008-2009
May 15	Baccalaureate/Commencement 2009

## Tuition Refund Policy

This is primarily of concern to part-time students. You can find the refund policy at <http://www.udmercy.edu/weblink/registrar/faqs/index.htm>. Note that it falls off very fast. No refund is available after two weeks..

## Disability Services:

The University complies with the Americans with Disabilities Act. Students who seek an accommodation should contact the Director of UAS/Disability Support Services in the Learning Center. More information is available at <http://www.udmercy.edu/uas/disability-support/index.htm>

## Ground Rules For Programming Assignments

1. Source Language: will be C++. As mentioned on the syllabus, Microsoft [Visual C++ Version C++ 6.0/Net](#) is the recommended compilers for the course.

I will give you a header file `443-rmc1.h` which will detect which version you are using and include the proper header files. If you are using version 6.0, it will include the pre-standard files. If using the more recent software, it will use the *namespace std* and include the corresponding headers.

The header file will include tools that allow you to date/time-stamp your output. These are based upon the *Microsoft Foundation Classes* (MFC). You will need to adjust the Project Settings to include the MFC classes.

2. Groups: You may work alone. However, on some of the projects you may also work with a partner. Requirements for individual projects will be a subset of those for the group projects.
3. Hand-in: For all projects, students should hand in a copy of their source files and output from a sample run of the program, using any test data that have been provided. In addition, the Output should always be date-stamped by the computer, indicating the date on which the output was produced.

I will also ask you to submit an electronic copy of the program. I will make provisions for you to submit through Blackboard. You should submit a ZIP file containing all your CPP and H files, any output files. I would also appreciate it if you would include an executable (EXE) file.

4. Style: Style is an important component of programming. Programs should employ meaningful identifiers and informative comments. Capitalization may be used provided it is in a consistent, coherent manner. (You may also choose to follow the text and use exclusively lower case identifiers with underscore characters to enhance readability.)

Unlike programs written in more self-documenting language like Pascal, CL programs tend to be cryptic and difficult to understand. Students should make judicious use of informative comments in their projects. Note that comments that merely repeat the meaning of an instruction are not informative. Comments should explain the purpose and functioning of code, not merely repeat it. (Some of the class handouts and text examples will have comments targeted at students learning the language for the first time. These comments would, of course, be inappropriate for other programs and for the homework assignments.)



Each program should certainly include a comment near the beginning giving the name of the project, the author, and his/her section number. Some explanation of each function is usually appropriate

5. Academic Integrity Students are expected to conform to a high standard of honesty and integrity in this course. Copying the work of someone else and other forms of cheating are strictly prohibited. Permitting or tolerating such behavior is also prohibited. The minimum penalty for any offense is a 0 on that assignment. The culprits may be subject to additional sanctions, up to and including expulsion from school for serious offenses, as prescribed by the University Catalog and the Engineering Science Student Handbook. For the group programming assignments, you obviously are free to cooperate with members of your own team. But the sharing of code between teams is explicitly forbidden.