



**TROY STATE UNIVERSITY  
CS 3323 DATA STRUCTURES  
COURSE SYLLABUS**

**PREREQUISITES:** *CS 2255 and MTH 2215.*

**INSTRUCTOR INFORMATION:**

**STUDENT EXPECTATION STATEMENT:**

It is assumed the student entering this course will already have a fundamental understanding of C++.

If you have not taken courses in C++ do not continue this class. You cannot successfully complete the course without the background knowledge given in the introductory and intermediate courses. You will be using algorithms you learned in Discrete Math to solve some of the problems given in the course.

Students are expected to be aware of the software life cycle and the importance of adequately and correctly documenting code. Students should be past the point of understanding how to take a program from pseudo-code through implementation (e.g. problem solving and implementing programs). This class will assume the student knows how to create and apply classes, objects and templates.

**COURSE DESCRIPTION:**

A course in fundamental data structures concepts and alternative techniques for solving real-world problems in computer science. Concepts and application covered include analysis of data representation and associated algorithms, including linked lists, queues, stacks, arrays, graphs, trees, searching, sorting, string matching, and the application of recursive techniques. The course will place an emphasis on the implementation of various algorithms and data structures. *Prerequisites: CS 2260 and MTH 2215.*

**Purpose:** To provide the student with hands on experience in implementing data structures and algorithms using C++.

**COURSE OBJECTIVES:**

Upon completion of this course, you should be capable of discussing and/or applying C++ programming methods in the following areas:

- Analysis of algorithms using Big-O notation
- Complexity analysis
- Linked lists
- Queues
- Stacks
- Recursion
- Trees
- Heaps
- Searching
- Sorting
- Hashing

### STUDENT OUTCOMES:

Upon completion of the course, the student will be able to:

- Critically analyze algorithms using the Big-O notation for algorithm comparison.
- Demonstrate a working knowledge of basic and factual material on data structure issues Standard Template Libraries, Abstract Data Types (ADT) and Object Oriented Design.
- Demonstrate a working knowledge of ADT major functions in an object-oriented programming environment.
- Demonstrate a working knowledge of arrays, vectors, lists, stacks, queues, linked lists, sorting techniques, trees structures, graphs, and hashing.
- Demonstrate improvement in their analytical and communication skills.
- Demonstrate all of the above via programming assignments and **interactive** discussions.

Students will demonstrate the above through their performance on the weekly quizzes, programming assignments, and class participation.

### SCHEDULE (TENTATIVE TO CHANGE)

Schedule	Topics	Chapters
Week 1	Review of C++ and Object Oriented Programming	Chapter 1 Appendix B
Week 2	Analysis of algorithms using Big-O notation	Chapter 2 Appendix A Appendix C
Week 3	Linked lists	Chapter 3

	And Arrays part I	
Week 4	Linked lists And Arrays part II	Chapter 3
Week 5	Stacks and Queues	Chapter 4
Week 6	Binary Trees, General Trees	Chapter 6
Week 7	Midterm	
Week 8	Binary Search Tree	Chapter 6
Week 9	AVL	Chapter 7
Week 10	Graphs part I	Chapter 8
Week 11	Graphs part II	Chapter 8
Week 12	Graphs part III	Chapter 8
Week 13	Sorting And Heaps	Chapter 9
Week 14	Hashing	Chapter 10
Week 15	Final Exam	

### **BREAKOUT OF HOW GRADES WILL BE DETERMINED:**

#### **Course Grade Components Approximate % of Grade**

Attendance	10
Exercises and Project	35
Midterm	25
Final exam	30

The overall course grade is based upon the student's performance on the weekly assignments, class participation, and quizzes.

### **ASSIGNMENT OF GRADES:**

Grades will be assigned according to point system on the following scale:

A	90 - 100
B	80 - 89
C	70 – 79
D	60 – 69
F	59 and below

### **HONESTY AND PLAGIARISM**

The awarding of a university degree attests that an individual has demonstrated mastery of a significant body of knowledge and skills of substantive value to society. Any type of dishonesty in securing those credentials therefore invites serious sanctions, up to and including suspension and expulsion (see Standard of Conduct in each TROY *Catalog*). Examples of dishonesty include actual or attempted cheating, plagiarism\*, or knowingly furnishing false information to any university employee.

\*Plagiarism is defined as submitting anything for credit in one course that has already been submitted for credit in another course, or copying any part of someone else's intellectual work – their ideas and/or words – published or unpublished, including that of other students, and portraying it as one's own. Proper quoting, using strict MLA formatting, is required, as described by the instructor. All students are required to read the material presented at: <http://troy.troy.edu/writingcenter/research.html>

### **TEXTBOOK(S) AND/OR OTHER MATERIALS NEEDED:**

- "Data Structures and Algorithms in C++, Third Edition", by Adam Drozdeck, Boston Massachusetts: Thomson Course Technology, **ISBN# 0-534-49182-0**
- Required Text: **ADTs, Data Structures, and Problem Solving with C++** by Nyhoff, Larry 2<sup>nd</sup> Edition, Prentice Hall, Inc. ISBN: 0-13-140909-3. You can obtain an electronic copy of the text at <http://www.safarix.com>.