Syllabus for
CSC 255—Data Structures
3 Credit hours
Fall 1999

I. COURSE DESCRIPTION

A study of the design of structures for representing information and the design of algorithms for manipulating that information. Expertise in the design of structures is developed through consideration of abstract structures in specific programming languages. Expertise in the design of algorithms is developed by solving problems in dynamic storage allocation, garbage collection, searching, and sorting. Programming projects throughout the course provide a synthesis experience in which the student designs data structures and algorithms to solve a given problem.

II. COURSE GOALS

The goals of this course include the following:

A. To train the student in the art of information representation and manipulation within a digital computer.

B. To give the student experience in the design of algorithms based on the principles of structured programming.

C. To enhance the computing maturity of the student and hence facilitate the transition from the perception of computing as a technical skill to an appreciation of the science of computing.

D. To prepare the student for employment in the computing industry.

E. To prepare the student for graduate work in computer science.

III. COURSE OBJECTIVES

A. Terminal Objective
As a result of successfully completing this course, the student will understand basic data structuring techniques and the basic principles of algorithm design as evidenced by the ability to
design and implement programs that solve typical data processing problems.

B. Unit Objectives

1. UNIT I The student will be able to write the definition, draw the logical diagram, design
a storage structure, and write manipulation algorithms for the array, stack, queue, deque,
ring, and list.

2. UNIT II The student will be able to write the definition, draw the logical diagram, design a storage structure, and write manipulation algorithms for tree and graph structures. Furthermore, the student will learn how to apply data structures to the problems of dynamic storage allocation and garbage collection as evidenced by the ability to design appropriate data structures and write algorithms based on those data structures for solving the problems in each of the areas.

3. UNIT III The student will be able to establish data structures and algorithms to accomplish the tasks of searching and sorting. Also, the student will be able to define the meaning of abstract data type and develop a notation for employing the concept in a programming language.

IV. TEXTBOOK

A. Required Textbook
Dale, Nell. C++ Plus Data Structures. Boston: Jones and Bartlett, 1999.

V. COURSE PROCEDURES

A. Course Prerequisite CSC 111—Introduction to Computing. Students must have experience in programming with a higher level language, such as Pascal or C/C++.

B. Evaluation Procedures
1. Homework—Homework assignments and programming problems will be given regularly in class. Details of specific requirements will be given at that time.
2. Final Grade—The final grade will be based approximately 20% on homework, 30% on programming problems, 30% on exams, and 20% on the final examination.

C. University Policies
1. "Consistent with Section 504 of The Rehabilitation Act of 1973, and the Americans With Disabilities Act, ORU ensures that no **"qualified individual"** will be denied reasonable accommodation in the form of modification of policies, practices and/or procedures." Students with disabilities who wish to access services should contact their professor(s) and the Office of Disability Services at Ext. 7355 to initiate the process.
2. Each student who uses the computer is given a user-name, password, and access to certain computer resources. These limited resources and privileges are given to allow students to perform course assignments. Abuse of these privileges will result in their curtailment. Students should note that the contents of computer directories are subject to review by instructors and the computer administrative staff.
3. A student taking a late exam for an unauthorized absence will be charged a late exam fee.
4. Students and faculty in the Department of Mathematics and Computer Science adhere to all laws addressing the ethical use of others' material, whether it be in the form of print, video, multimedia, or computer software.

VI. ATTENDANCE POLICY

A. Attendance at each class or laboratory is mandatory at Oral Roberts University.

B. Double cuts will be assessed for absences that immediately precede or follow holidays or breaks.

C. In keeping with both the spirit and the letter of the ORU Honor Code, the Mathematics/Computer Science Department expects your attendance at each class period. A LOSS OF POINTS MAY OCCUR AS A RESULT OF ABSENCES FROM CLASS.

D. An administratively excused absence (pre-approved by the Academic Dean) allows the assignment to be made up without penalty; however, the student must take the responsibility for furnishing the excuse to the instructor, who will ascertain when the makeup work is expected.

E. All other absences, regardless of reason, are unexcused. Some events, beyond your own control, can occur which prevent your attendance. The instructor has the right to evaluate such cases on an individual basis to ascertain if a penalty will be assessed. It remains the responsibility of the student to check with the instructor immediately upon return to class to determine the consequences of said absence.

F. In cases where an exam is missed and must be administered later, a late-exam fee will be assessed.

G. It is the right of the instructor to devise methods to achieve good attendance and punctuality. These methods may include pop quizzes, which cannot be made up, bonus points for perfect attendance, loss of points for any absences, or other procedures to encourage attendance.

VII. COURSE CALENDAR

Unit	Lesson	Topic
I	1	Introduction
	2-4	Structured programming
	5-12	Software Engineering and Data Design
	13	EXAM I
II	14-18	Lists
	19-23	Stacks and Queues
	24-25	Recursion
	26	EXAM II
III	27-32	Trees and Graphs
	33-34	Sorting
	35-37	Searching
	38	EXAM III

39-44	Group projects
45	FINAL EXAM

Dr. Kenneth Preston	CSC 255	Data Structures	Math/CSC
Name of Instructor	Course #	Title of Course	Name of Department

MISSION

The lifestyle at ORU is rooted in the word "Wholeness." ORU seeks to educate the whole person, with balanced emphasis placed on the development of mind, spirit, and body.

GENERAL OUTCOMES

1. Spiritual Development
2. Physical Development
3. Communication
4. Analysis
5. Problem Solving
6. Valuing in Decision Making
7. Social Interaction
8. Global Perspectives
9. Effective Citizenship
10. Aesthetic Responsiveness

MAJOR OUTCOMES

Achieves competence in the use of computer hardware and software, and is able to effectively communicate with those seeking to utilize computers in solving problems.

Develops the ability to describe the process by which a problem can be solved.

Analyzes problems to determine how theoretical patterns can be adapted to formulate practical solutions.

Gains experience in working with groups of people to develop computerized systems.

COURSE OUTCOMES

Design and implement data structures and the algorithms that accompany them.

Write and analyze algorithms that accomplish the tasks of sorting, searching, and dynamic storage allocation.

Discuss the principles of abstract data types and object oriented programming.

ASSESSMENT OF COURSE GOALS

STIMULI

Homework

Programs

Group Projects

Exams

CRITERIA

Final grade is based on the following:

Homework	20%
Programs	30%
Exams	30%
Final Exam	20%