DATA STRUCTURES
CSIT 1400 (formerly CST 1400)

Class Hours: 3.0
Laboratory Hours: 3.0
Credit Hours: 4.0
Date Revised: Spring 03

Catalog Course Description:
Advanced problem solving and algorithm development, structured programming, data structures and applications, I/O techniques, lists, queues, trees, algorithms, and files. Program development using UNIX operating system. This course is intended for university parallel students.

Entry Level Standards:
The student is expected to be proficient in C programming components taught in CSIT 1020. These include functions, arrays, string handling, argument passing, indirect addressing and elementary file I/O. The student is also expected to have a working knowledge of the Unix operating system, a Unix-based editor such as vi or emacs and C program development in the Unix environment. The student must have math, writing, verbal and English language skills at the college level.

Prerequisite:
CSIT 1020 or department approval

Textbook(s) and Other Reference Materials Basic to the Course:
Weiss, Mark Allen; Data Structures and Algorithm Analysis in C, 2nd Edition; Addison-Wesley; 1997

I. Week/Unit/Topic Basis:

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction, Overview of Algorithm Complexity, Review of C: Arrays, Strings, Pointers, Pointer Arithmetic, Indirection, Double Indirection, Functions, Arguments and Scope of Variables, Prototypes, Program Structure</td>
</tr>
<tr>
<td>2</td>
<td>Review of C</td>
</tr>
<tr>
<td>3</td>
<td>Structures, Typedef, Dynamic Memory Allocation</td>
</tr>
<tr>
<td>4</td>
<td>Lists, Stacks, Queues, Static and Dynamic Lists</td>
</tr>
<tr>
<td>5</td>
<td>Record I/O, Command Line Arguments, Exam I</td>
</tr>
<tr>
<td>6</td>
<td>Binary trees, Tree Traversals, Insertions and Deletions, AVL Trees</td>
</tr>
<tr>
<td>7</td>
<td>B Trees</td>
</tr>
</tbody>
</table>
II. Course Objectives*:

A. Demonstrate proficiency in the C and C++ programming language. III VI VII IX XI

B. Demonstrate use of advanced C programming statements and able to use these statements in writing a large program. III VI VII IX XI

C. Demonstrate knowledge of data abstraction, specification, refinement and implementation, understanding of specific structures such as lists, stacks, queues, linked-lists, hash tables and binary trees. III IV XI XII

D. Demonstrate use of various searching and sorting methods and select most efficient algorithm. III V XI XII

E. Demonstrate use of various data structures in writing a large program with C. III V X XI XII

F. Write well-structured programming code using divide-and-conquer method. II III V VI VII IX X XI XII

G. Use recursive techniques to solve problems. V, VI, IX

*Roman numerals after course objectives reference goals of the CSIT program.

III. Instructional Processes*:

Students will:

1. Create a complex software package which implements multiple data structures. Communication Outcome, Problem Solving and Decision Making Outcome, Technological Literacy Outcome, Information Literacy Outcome, Personal Development Outcome, Transitional Strategy, Active Learning

2. Examine and implement algorithms that are efficient and reliable. Problem Solving and Decision Making Outcome, Technological Literacy Outcome, Information Literacy Outcome, Personal Development Outcome, Transitional Strategy, Active Learning

3. Use professional tools to produce software components and documentation. Technological Literacy Outcome, Personal Development Outcome, Transitional Strategy, Active Learning
4. Participate in a software development team. *Communication Outcome, Problem Solving and Decision Making Outcome, Personal Development Outcome, Transitional Strategy, Active Learning Strategy*

5. Participate in a peer review of term projects. *Problem Solving and Decision Making Outcome, Communication Outcome, Transitional Strategy, Active Learning Strategy*

6. Use professionally accepted methods and materials in completion of applications. *Technological Literacy Outcome, Personal Development Outcome, Transitional Strategy, Active Learning*

7. Practice elements of the work ethic such as punctuality, professionalism, dependability, cooperation, and contribution. *Personal Development Outcome*

*Strategies and outcomes listed after instructional processes reference Pellissippi State’s goals for strengthening general education knowledge and skills, connecting coursework to experiences beyond the classroom, and encouraging students to take active and responsible roles in the educational process.

**IV. Expectations for Student Performance***:

Upon successful completion of this course, the student should be able to:

1. Learn the syntax and semantics of C and C++ programming language. A

2. Utilize advanced C programming statements in large programs. B

3. Understand simple data types, arrays, structures and unions. B

4. Understand implementation of abstract data structures via pointers. B, C

5. Understand links, stacks, queues, linked-list and binary tree searching. C

6. Understand trees and tree traversal. C

7. Understand recursive functions. C, D

8. Understand various sorting and searching techniques. D

9. Understand graphs and graph traversal. D

10. Understand hashing techniques. D

11. Understand heaps and their applications. D

12. Write a large program using various data structures. E, F

13. Use recursion as an alternative to linear solutions. A, B, C, G

14. Use *make files* to manage projects. F

*Letters after performance expectations reference the course objectives listed above.*

**V. Evaluation:**

**A. Testing Procedures:**

A minimum of three tests is recommended. Tests will cover material presented in class. Tests
are not to be missed without a valid excuse.

B. Laboratory Expectations:

Laboratory programming assignments will be given for each laboratory period. Lab assignments will count 200 points. A late penalty will be assessed on overdue assignments.

C. Field Work:

N/A

D. Other Evaluation Methods:

Class participation, quizzes and homework will also comprise the final grade for the course.

E. Grading Scale:

93 – 100 A  
88 – 92 B+ 
83 – 87 B  
78 – 82 C+  
73 – 77 C  
65 – 72 D  
Below 65 F

VI. Policies:

A. Attendance Policy:

Pellissippi State Technical Community College expects students to attend all scheduled instructional activities. As a minimum, students in all courses must be present for at least 75 percent of their scheduled class and laboratory meetings in order to receive credit for the course (Pellissippi State Catalog).

B. Academic Dishonesty:

Plagiarism, cheating and other forms of academic dishonesty are prohibited. A student guilty of academic misconduct, either directly or indirectly through participation or assistance, is immediately responsible to the instructor of the class. In addition to other possible disciplinary sanctions that may be imposed through the regular Pellissippi State procedures as a result of academic misconduct, the instructor has the authority to assign an F or a zero for the exercise or examination or to assign an F in the course.