

**Master Course Syllabus**  
School of Engineering and Computer Science  
Washington State University Vancouver

**CS 122**

**Data Structures**

4 Semester Hours

(3 lecture hours, 3 laboratory hours)

**Catalog Description**

Advanced programming techniques: data structures, recursion, sorting and searching, and basics of algorithm analysis.

**Prerequisite Courses**

CS 121 – Program Design and Development

**Prerequisite Topics**

- Imperative programming language fundamental concepts
- Proficiency in fundamental C language constructs

**Measured Course Outcomes**

Students taking this course will:

1. Design, code, test and debug recursive functions and contrast them with iterative strategies. (*Contributes to performance criterion A-2*)
2. Select and implement appropriate fundamental data structures (including dynamically allocated linked lists and trees) and algorithms to solve problems and use them in programs. (*Contributes to performance criterion E-1*)
3. Analyze and contrast the space and time complexity of several fundamental sorting and searching algorithms. (*Contributes to performance criterion A-3*)

**Required Textbooks**

Algorithms in C, Third Edition, Parts 1-4: Fundamentals, Sorting, Searching, and Strings, Robert Sedgewick, 1998, ISBN 0-201-31452-5.

or

Programming Abstractions in C: A Second Course in Computer Science, Eric S. Roberts, 1998, Addison Wesley.

## Reference Material

C How to Program, Deitel and Deitel, Prentice Hall, 2001.

C: The Complete Reference, Fourth Edition, Herbert Schildt, McGraw Hill.

## Major Topics Covered in the Course

1. Memory allocation and de-allocation, memory addressing, pointers
2. Data structures including strings, linked lists, stacks, queues, binary search trees, and hash tables
3. Recursion
4. Search algorithms (including linear and binary)
5. Sorting algorithms (including merge sort, quick sort and heap sort)
6. Simple complexity analysis of algorithms

## Laboratory Projects

All programming projects and assignments are to be derived and developed by students individually. In this course, students do not perform assignments as members of teams.

---

Programming Project Area	Weeks
--------------------------	-------

---

## CSAB Category Content

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	2	0	Computer Organization and Architecture	0	0
Algorithm & Software Design	2	0	Concepts of Programming Languages	0	0

### **Oral and Written Communications**

There are no significant oral or written communications required in this course. Virtually all assignments consist of writing computer programs.

### **Social and Ethical Issues**

This course contains no significant coverage of social and ethical issues beyond proscriptions against plagiarism and cheating.

### **Theoretical Content**

Topic	Hours
Algorithm complexity	4

### **Problem Analysis**

The instructor performs analysis of representative problems in class. All student programming assignments require the student to analyze problem requirements. The instructor analyzes problem solutions (both his own and student's) in class.

### **Solution Design**

This course requires the student to craft 10-12 correctly functioning computer programs. The requirements for each program will necessitate that the student comprehend programming language concepts and use them to design and implement programs in C. These programs range from 30-50 lines of code at the beginning of the semester, to a final assignment whose design usually entails ~400 lines of program code.

### **CC2001**

This course provides coverage of topics in the following areas (hours listed are minimums):

PF1. Fundamental programming constructs [core]	2
PF2. Algorithms and problem solving [core]	3
PF3. Fundamental Data Structures [core]	11
PF4. Recursion [core]	5
AL1. Basic algorithm analysis [core]	1
AL3. Fundamental computing algorithms [core]	7

---

Course Coordinator: Dick Lang  
Last Updated: September 8, 2005      Approved  
Syllabus Version Number: 1.2