Master Course Syllabus

School of Engineering and Computer Science
Washington State University Vancouver

CS 223

Advanced Data Structures

3 Semester Hours

Catalog Description

Advanced data structures, object oriented programming concepts, concurrency, and program design principles.

Prerequisite Courses

CS 122 – Data Structures

CS 216 – Discrete Structures

Prerequisite Topics

- Proficiency with the C programming language, including the use of pointers
- Fundamental data structures
- Basic sorting and searching algorithms
- Graph traversal methods

Measured Course Outcomes

Students taking this course will:

1. Apply and implement advanced data structures, such as B-trees, multi-way trees, balanced trees, heaps, priority queues, to solve computational problems *(contributes to performance criterion **E-1**)*.
2. Design, code, test and debug simple programs in an object-oriented language demonstrating the use of polymorphism, inheritance and encapsulation techniques *(contributes to performance criterion **K-2**)*.
3. Determine the time and space complexity of algorithms, including recursively defined algorithms *(contributes to performance criterion **A-3**)*.

Required Textbooks

One of the following:

- Data Structures and Algorithms in Java, 2nd Edition, Robert Lafore, Sams, 2002, ISBN 0672324539.

- Algorithms in Java, Robert Sedgewick, Addison-Wesley, 3rd Edition, 2003, ISBN 0201775786.
- Algorithms in C++ Third Edition, Parts 1-4: Fundamentals, Data Structures, Sorting, Searching, Robert Sedgewick, 1999, ISBN 0-201-35088-2, Addison Wesley.
- Data Structures and Algorithm Analysis in C++, 2/e, Mark Weiss, 1999 ISBN, 0-201-36122-1, Addison Wesley.

Reference Material

None specified.

Major Topics Covered in the Course

1. Abstract data types
2. Object oriented design and programming concepts including inheritance, polymorphism, modularity, encapsulation, and overloading
3. Advanced data structures including balanced binary search trees, B-trees, multi-way trees, and hash tables
4. Associative arrays, graphs, graph traversal algorithms, sets and relations
5. Memory management and memory managers
6. Concepts of concurrency

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
Search tree implementation	2
Hash table or associative array programming	2
Graph traversal	2
ADT, inheritance, polymorphism usage	2
Perform depth first and breadth first search	2

CSAB Category Content

	FUNDAMENTAL	ADVANCED		FUNDAMENTAL	ADVANCED
Data Structures	1	0	Computer Organization and Architecture	0	0
Algorithm & Software	1	0	Concepts of Programming	1	0

Design				Languages		

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 the usual proscriptions against plagiarism and cheating.

Theoretical Content

Topic	Hours
Concepts of concurrency and synchronization	3

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 8-12 correctly functioning computer programs. The requirements for each program will necessitate that the student comprehend and apply mathematical knowledge and lecture material to design and implement programs in C, C++ and/or Java. These programs range from 50-200 lines of code, increasing in size and complexity towards the end of the semester.

CC2001

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

AL1. Basic Algorithmic Analysis [core]	3
AL2. Algorithmic strategies [core]	2
AL3. Fundamental computing algorithms [core]	5
DS5. Graphs and trees [core]	2
PL4. Declarations and types [core]	3
PL5. Abstraction mechanisms [core]	3
PL6. Object-oriented programming [core]	10