

SU DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE
SYLLABUS (Tentative)
COSC 220 Computer Science II

Description: A study of the design and implementation of abstract data types and algorithms using an object-oriented approach and standard class library. Attention will be paid to the introduction of data structures such as linked lists, vectors, stacks, queues, priority queues, lists, trees, etc.; searching and sorting algorithms and their runtime analysis. C++ is the teaching language. Three one-hour lectures and one two-hour lab per week.

Prerequisite: Computer Programming (COSC 120) or equivalent with a grade of C or better and Discrete Mathematics (MATH 210) or equivalent with a grade of C or better. MATH 210 may be taken concurrently.

Required Text: “Starting Out with C++,” by Tony Gaddis; Pearson/Addison Wesley, 9th Edition. ISBN: 978034498379.

	<i>Week</i>
<i>Pointers & Arrays</i>	<i>1.0</i>
Review pointers, passing pointers as parameters, relation between pointers and arrays, dynamic memory allocation, arrays of pointer types	
<i>Advanced Recursion</i>	<i>1.0</i>
Recursive algorithms and functions	
<i>Linked Lists</i>	<i>2.0</i>
Introduce singly-linked and doubly-linked lists and their manipulation	
<i>Data Structures</i>	<i>2.0</i>
Introduce vectors, stacks, queues (include priority queues) and their manipulation through their APIs	
<i>Data Structure Implementation</i>	<i>1.5</i>
Discuss implementation of vector, stack, queue, list using pointer-based array and/or linked lists	
<i>Sorting Algorithms</i>	<i>1.5</i>
Introduce algorithms for insertion sort, mergesort, quicksort, radix sort, etc. with arrays	
<i>Algorithm Efficiency Analysis</i>	<i>1.0</i>
Introduce asymptotic notations (big-O, big-Ω, big-Θ) and basic related theorems, perform runtime analysis on searching and sorting algorithms	
<i>Advanced Concepts in Object-oriented Programming</i>	<i>2.0</i>
Introduce inheritance, polymorphism, abstract classes, virtual functions, static and dynamic binding	
<i>Optional Topics</i>	<i>1.0</i>
Introduce trees, hash tables, heaps and other data structures	
<i>Test</i>	<i>1.0</i>
	<i>14.0</i>

EVALUATION

Homework, labs, projects, class participation, presentation: 30-50%

Tests, final exam, and quizzes: 50-70 %

NOTE: ONCE A STUDENT HAS RECEIVED CREDIT, INCLUDING TRANSFER CREDIT, FOR A COURSE, CREDIT MAY NOT BE RECEIVED FOR ANY COURSE WITH MATERIAL THAT IS EQUIVALENT TO IT OR IS A PREREQUISITE FOR IT.