CS 2020 : INTERMEDIATE PROGRAMMING

Semester Hours:	3.0	*Contact Hours:* 3
Coordinator:	Ronald Conway	
Text:	Intermediate Programming with zyBooks & zyLabs	
Author(s):	VAHID & LYSECKY	
Year:	2022	

SPECIFIC COURSE INFORMATION

Catalog Description:

Introduction to object-oriented programming techniques. Constructors, destructors, operator overloading. Inheritance and polymorphism. Elementary data structures including linked lists. Dynamic storage allocation concepts. Prerequisite: Corequisite of MATH 1260 or MATH 1280 or MATH 1300 (Precalculus) or higher and grade of C or better in CS 2010. Approved for distance education.

Course type: **REQUIRED**

SPECIFIC COURSE GOALS

- I can understand and can implement search and sorting algorithms.
- I can implement programs using arrays and linked lists.
- I can use dynamic memory techniques in implementing programming design.
- I can use fundamental object-oriented programming techniques, including encapsulation, inheritance, polymorphism, and virtual functions.

COMPUTER SCIENCE STUDENT OUTCOMES ADDRESSED BY THIS COURSE

- CS 1 Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions
- CS 2 Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline
- CS 6 Apply computer science theory and software development fundamentals to produce computing-based solutions

SOFTWARE ENGINEERING STUDENT OUTCOMES ADDRESSED BY THIS COURSE

- SE 1 An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

LIST OF TOPICS COVERED

- Struct and Classes (2 weeks = 14%)
    - Grouping data: struct
    - Structs and functions
    - Objects: Introduction
        - ADTs
        - Preconditions, Postconditions, and Class Invariants
    - Using a class
    - Mutators, accessors, and private helpers
    - Separate files for classes
- More Classes (new, delete) (2.5 week = 18%)
    - Initialization and constructors
    - Classes and vectors/classes
    - Unit testing (classes)
    - Constructor overloading
    - Operator overloading
- Pointer (2 weeks = 14%)
    - Pointer basics
    - Pointer to Arrays
    - Pointer Arithmetic
    - Operators: new, delete, and ->
    - Memory regions: Heap/Stack
- Vectors/Dynamic arrays (2 weeks = 14%)
    - Iterating through vectors
    - Multiple vectors
    - Vector resize

    - Vector push_back
- Linked Lists (2.5 weeks = 18%)
    - A first linked list
    - Memory leaks
    - Destructors
    - Rule of three
- Introduction to inheritance and polymorphism (2 weeks = 14%)
    - Derived classes
    - Access by members of derived classes
    - Overriding member functions
    - Polymorphism and virtual member functions
    - Abstract classes
    - Is-a versus has-a relationships
    - UML
- Recursion (direct/linear & binary) (1 week = 7%)
    - Recursive functions
    - Recursive algorithm: Search
    - Creating a recursive function
    - Stack overflow
- Function templates (.5 weeks = 5%)
    - Function templates
    - Class templates

COMPUTER SECURITY TOPICS

Faculty who recently offered CS 2020 have discussed and identified a list of topics related to computer security in this course. Below is a list for instructors to incorporate. (*) indicates topics that are mandatory.

Security Topic	Description	Textbook Reference[1]	Estimated Class Hours
*Bounds Checking	Pointer manipulations, vector access – index and pointers	Module 10 Module 13	<1
*Principle of Least Privilege	Default private struct – default public; other access modifiers. Class access modifiers	Module 11	<1
*Obfuscation	Obscures intended meaning; for example, operator overloading	Module 12	<1
*Access Control	Inheritance, polymorphism, lack of security with friendship	Module 15	1

[1] zyBooks: CS2020: Intermediate Programming.