CS 2020 : INTERMEDIATE PROGRAMMING

<table>
<thead>
<tr>
<th>Semester Hours:</th>
<th>3.0</th>
<th>Contact Hours:</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinator:</td>
<td>Ronald Conway</td>
<td>Text:</td>
<td>Intermediate Programming with zyBooks &amp; zyLabs</td>
</tr>
<tr>
<td>Author(s):</td>
<td>VAHID &amp; LYSECKY</td>
<td>Year:</td>
<td>2022</td>
</tr>
</tbody>
</table>

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%)
  o Grouping data: struct
  o Structs and functions
  o Objects: Introduction
    ▪ ADTs
    ▪ Preconditions, Postconditions, and Class Invariants
  o Using a class
  o Mutators, accessors, and private helpers
  o Separate files for classes

• More Classes (new, delete) (2.5 week = 18%)
  o Initialization and constructors
  o Classes and vectors/classes
  o Unit testing (classes)
  o Constructor overloading
  o Operator overloading

• Pointer (2 weeks = 14%)
  o Pointer basics
  o Pointer to Arrays
  o Pointer Arithmetic
  o Operators: new, delete, and ->
  o Memory regions: Heap/Stack

• Vectors/Dynamic arrays (2 weeks = 14%)
  o 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
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.

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