Course Description

Examines fundamental data structures and analyzes algorithms. Covers abstract data types and essential data structures such as arrays, stacks, queues, linked lists, and trees; introduces searching and sorting algorithms and algorithm analysis. Lecture 4 hours per week.

General Course Purpose

Examines data structures and analyzes algorithms. Covers data structures (including sets, strings, stacks, queues, arrays, records, files, linked lists, and trees), abstract data types, algorithm analysis (including searching and sorting methods), and objects.

Course Prerequisites/Corequisites

Prerequisite: Prerequisite: CSC 202 and MTH 263. Corequisite: MTH 264 or division approval.

Course Objectives

Upon completion of this course, the student will be able to:

A. Analyze Problems
   B. Develop Algorithms in an Object Oriented Environment

Major Topics to be Included

I. Data abstraction and object-oriented programming (review)
   A. dynamic allocation
   B. overloading functions and operators
   C. constructors and destructors

II. Recursion
   A. analysis
   B. applications

III. ADT implementations (static and dynamic) and applications
   A. linked lists (singly, doubly, circular)
   B. stacks
   C. queues
   D. trees and tree traversal

IV. Searching
   A. analyzing and comparing searching methods
   B. relationships between methods and data structures

V. Sorting
   A. examples of order n^2 and order n log( n) sorts
   B. analyzing and comparing sorting methods
VI. Optional other topics
   A. hashing
   B. tables
   C. file structures
   D. sets
   E. Templates

5. Suggested Time Allocation per Topic

In order to standardize the core topics of CSC 202 so that a course taught at one campus is equivalent to the same
course taught at another campus, the following student contact hours per topic are recommended. Of course, the
topics cannot be followed sequentially. Many topics are taught best as an integrated whole, often revisiting the topic
several times, each time at a higher level. There are normally 64 student-contact-hours per semester for a four-unit
course. The last category, *Other/Enhance*, leaves ample time for an instructor to tailor the course to special needs or
resources.

<table>
<thead>
<tr>
<th>Ref</th>
<th>Topic</th>
<th>Hours</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>Data abstraction and object-oriented programming</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>II</td>
<td>Concept of Recursion</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>III A</td>
<td>Linked lists/pointers</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>III B</td>
<td>Stacks</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>III C</td>
<td>Queues</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td>III D</td>
<td>Trees</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>IV</td>
<td>Searching</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td>V</td>
<td>Sorting</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>VI</td>
<td><em>Other</em> optional content or <em>Enhance</em> the above</td>
<td>8</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>Exams and quizzes</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td><strong>Total</strong></td>
<td><strong>64</strong></td>
<td><strong>100</strong></td>
</tr>
</tbody>
</table>