<table>
<thead>
<tr>
<th><strong>Course</strong></th>
<th>ECE 36800 - Data Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type of Course</strong></td>
<td>Required for CmpE Program, Elective for EE Program</td>
</tr>
<tr>
<td><strong>Catalog Description</strong></td>
<td>Provides insight into the use of data structures. Topics include stacks, queues and lists, trees, graphs, sorting, searching, and hashing.</td>
</tr>
<tr>
<td><strong>Credits</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Contact Hours</strong></td>
<td>3</td>
</tr>
<tr>
<td><strong>Prerequisite Courses</strong></td>
<td>CS 22800</td>
</tr>
<tr>
<td><strong>Prerequisites by Topics</strong></td>
<td>Programming experience in C/C++. Experience in using software scripting tools and software testing tools.</td>
</tr>
<tr>
<td><strong>Course Objectives</strong></td>
<td>This course provides insight into the concept of data structures, its implementation and application. Covered topics include data structures of lists, stacks, queues, trees, and graphs. Associated algorithms of searching, sorting, hashing, and graph algorithms are also covered.</td>
</tr>
</tbody>
</table>
| **Course Outcomes**  | Students who successfully complete this course will have demonstrated:  
1. An understanding of abstract concepts and the principle of design using abstract data types (ADTs). (a, k)  
2. An understanding of how the abstract concepts can be made concrete by using C++ programming language. (a, b, k)  
3. An understanding of certain specific data structures (list, stack, queue, tree, and graph). (b, k)  
4. An understanding of associated algorithms to see how data structures design can be useful in problem solving. (b, k)  
5. An ability to analyze the time complexity of simple algorithms. (a)  
6. An ability to apply data structures and associated algorithms to solve engineering problems. (a, b, e, k) |
| **Lecture Topics**   | 1. Review on C++  
2. Principles of Object Oriented Programming in C++ |
3. Complexity analysis
4. Basic Abstract Data Types (ADTs)
5. Arrays and lists
6. Stacks and queues
7. Recursion
8. Trees
9. Search algorithms
10. Hashing
11. Sort algorithms
12. Graphs and graph algorithms

Computer Usage High
Laboratory Experience High
Design Experience High
Coordinator Chao Chen, Ph.D.
Date 31/3/11