1. **Course number and name:** CSCI 3230 Data Structures

2. **Credits and contact hours:** 3 credit, 3 contact

3. **Instructor’s or course coordinator’s name:** Youming Li, PhD

   a. **Other supplemental materials:** None

5. **Specific course information**
   a. **Brief description of the content of the course (Catalog Description)**
      Introduction to abstract data types such as lists, stacks, queues, and trees, and algorithm analysis.
      **Prerequisites:** A minimum grade of “C” in CSCI 1302 and MATH 2130
   b. **Indicate whether a required, elective, or selected elective course in the program**
      Required course for BS-CS

6. **Specific goals for the course**
   a. **Specific outcomes of instruction, ex. The student will be able to explain the significance of current research about a particular topic.**

<table>
<thead>
<tr>
<th>Course Learning Outcomes</th>
<th>Student Outcomes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understand $O$, $\Omega$, and $\Theta$ and time/space complexity</td>
<td>1a, 1i, 2a</td>
</tr>
<tr>
<td>Ability to understand the concepts of lists, stacks, and queues, and to implement them using arrays/linked structures</td>
<td>1a, 1b, 1c, 1i, 2a, 2b</td>
</tr>
<tr>
<td>Ability to understand, and to implement linear and binary searches</td>
<td>1a, 1b, 1c, 1i, 2a, 2b</td>
</tr>
<tr>
<td>Ability to understand, program, and apply $O(N^2)$, $O(N\log(N))$ and possible other sorts of lists</td>
<td>1a, 1b, 1c, 1i, 2a, 2b</td>
</tr>
<tr>
<td>Understand the concept of divide and conquer wrt algorithms</td>
<td>1a, 1b, 1c, 1i, 2a, 2b</td>
</tr>
<tr>
<td>Understand the concept of a hash function and its application in search</td>
<td>1a, 1b, 1c, 1i, 2a, 2b</td>
</tr>
<tr>
<td>Understand concepts in graphs: directed/undirected/ oriented, graphs and chains/ paths/cycles</td>
<td>1a, 1i, 2a</td>
</tr>
<tr>
<td>Understand the concepts of weakly and strongly connected graphs</td>
<td>1a, 1b, 1c, 1i, 2a, 2b</td>
</tr>
<tr>
<td>Understand the concept of a tree, a forest, a singly rooted tree, and a binary tree</td>
<td>1a, 1i, 2a</td>
</tr>
<tr>
<td>Understand the concept of ancestral relationships in trees</td>
<td>1a, 1i, 2a</td>
</tr>
</tbody>
</table>
Understand the concept of a binary search tree, and modify a binary tree to a binary search tree 1a, 1b, 1c, 1i, 2a, 2b

Ability to program preorder, inorder, and postorder binary tree traversals 1a, 1b, 1c, 1i, 2a, 2b

Understand the concept of self-balancing trees, AVL trees and optionally B-Trees and B*-Trees 1a, 1b, 1c, 1i, 2a, 2b

Understand heaps, its implementation and its application to sorting and priority queues 1a, 1b, 1c, 1i, 2a, 2b

Ability to understand and program depth-first and breadth-first searches 1a, 1i, 2a

Understand Dijkstra’s shortest path algorithm 1a, 1b, 1c, 1i, 2a, 2b

Ability to understand the concept of a minimal spanning tree and to find a minimal spanning trees for a graph 1a, 1b, 1c, 1i, 2a, 2b

Math outcomes (highlighted in green): 6 out of 17; approximately 35%, or 1 credit hour of math in CSCI 3230.

b. Student Outcomes:
   - 1a: An ability to apply knowledge of computing and mathematics appropriate to the discipline
   - 1b: An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
   - 1c: An ability to design, implement and evaluate a computer-based system, process, component, or program to meet desired needs
   - 1i: An ability to use current techniques, skills, and tools necessary for computing practice
   - 2a: An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
   - 2b: An ability to apply design and development principles in the construction of software systems of varying complexity

7. Brief list of topics to be covered
   - collections
   - lists (including linked lists, queues, stacks)
   - searching and sorting algorithms
   - trees (including binary search trees and heaps)
   - hashing
   - graphs and networks