In CSC111 you were initially introduced to primitive types (for example, int, float, char) and learned how to write simple programs using them. Next, you learned about objects and references to objects and then your programs reflected not only the data items in those objects, but also methods that manipulate them. Lastly, arrays were introduced as a container for multiple variables. In CSC112 you learned more about objects and pointers to objects, and ultimately your programs were a mixture of linked lists, arrays, primitive types and classes. However, in both CSC111 and CSC112 the overriding goal was to learn about object oriented programming in Java and in C++. In this course, the focus is not on programming, but we all will need to construct Java programs, but on the design, implementation and application of data structures and related algorithms. Data structures include classes, arrays, linked lists and much more.

Course Description from Wake Forest Bulletin, 2016-2017:

221. Data Structures and Algorithms I. (3h) Analysis, implementation, and application of abstract data structures such as lists, stacks, queues, trees, hash tables, heaps and graphs. Complexity analysis of algorithms that operate upon these structures. P—CSC 112. P or C—MTH117. (D)

Course Learning Objectives:
In this course, students will learn how an abstract data type is implemented as a data structure, how an abstract data type is applied in a problem, and how the time and space complexity of an algorithm is computed. Both abstract reasoning and practical implementation will be important tools in this course.

Course Topics:

Complexity
The complexity classes of O(), Omega(), and Theta() will be introduced and will be used throughout the course.

Lists
The data structures arrays and singly and doubly linked lists are studied as implementations of the abstract data type List. Particular attention is given to static and dynamically allocated arrays of dimension two and three. Singly linked lists (previously covered in CSC112) are compared to doubly linked lists.

Stacks and Queues
These abstract data types (previously covered in CSC112), and the associated data structures, are studied with special emphasis on the constant time complexities of push() and pop().

Binary Search Trees
Binary search trees are presented as a non-linear, order driven, abstract data type to facilitate searching. Special emphasis is given to the complexity of the operations of searching, inserting and deleting.

Heaps
These binary trees are presented as an efficient abstract data type for ordered data where the root is of paramount interest. As an example of its use, an algorithm such as find_max() is presented and analyzed.

Hash Tables
Hash tables, implemented directly and with chaining, are presented as a highly efficient abstract data type for storing and searching for data. Time complexities associated with find() are analyzed.

Graphs
The abstract data type of graph (the data version of a mathematical relation), are studied. Operations of insert, delete and find are discussed. Algorithms over graphs, such as minimal spanning tree, are presented, along with their complexity analyses.
Sorting
Two sorting algorithms, such as quick sort and heap sort, are implemented and analyzed.

Tools and Skills
The programming language used for implementation is Java. An appropriate and modern IDE, such as IntelliJ, will be used by the class to support program construction, compilation, linking and execution, including interactive debugging. At least one of the labs will incorporate external code that requires the modification of a program building tool, such as Cmake.

Labs
There must be at least five lab assignments based on applying these abstract data types. At least three of these assignments must involve at least partial implementation of a data structure. At least one of these must involve using the implementation of a data structure incorporating an existing library.

Honor System
Wake Forest is an academic community that subscribes to an honor system. By accepting membership in this community, each student assumes the obligation to be trustworthy in all pursuits. Violations may be referred to the Honor Council for investigation and determination of appropriate sanctions.

Special Needs
If you have a disability that may require an accommodation for taking this course, then please contact the Learning Assistance Center at (758-5929) within the first two weeks of the semester.

Plan in the Case of Campus Closing
Please note the following plan to be followed in the event that the Wake Forest campus is closed for an extended period of time and we are unable to have our regularly-scheduled class meetings.

Be sure to take your book, computer, and course notes home with you in the event that the campus is closed. We’ll continue with tests and programming assignments, communicating through the internet, email, and/or hard mail.

If internet access is available, assignments should be put into the appropriate Sakai assignment folder.

If the internet is down, I will mail your assignments to you in hard copy, and, by return address, you should mail back a flash drive containing the source code for the implemented program or a file containing the homework solutions.

In normal circumstances, please contact me through my campus email address or campus telephone number.
campus email: djjj@wfu.edu
campus telephone: 336-758-5535

Your course information, including a schedule of assignments, will be posted at on Sakai.
After leaving campus, you should regularly consult the schedule on Sakai for updates to the schedule.

Grading
Your grade will be computed from two in-class tests (40%), one final exam (25%), five lab assignments (30%), and homeworks related to the Zybook text (5%).

Tests & Final Exam
Test #1 -- Wednesday, February 22
Test #2 -- Monday, April 10
Final Exam -- Wednesday, May 3, 7-10pm