1. **Course number and name**  
   CptS 223: Advanced Data Structures in C++

2. **Credits and contact hours**  
   3 credits, 3 lecture hours

3. **Instructor’s or course coordinator’s name**  
   Aaron Crandall

4. **Textbook, title, author, and year**  

5. **Specific course information**  
   a. **Catalog description:** Advanced data structures, object oriented programming concepts, concurrency, and program design principles taught in C/C++ programming language.  
   b. **Prerequisites or corequisites:** CptS 122, MATH 216 (concurrent enrollment okay).

6. **Specific goals for the course**  
   By the end of the course, students will be able to  
   ● Analyze and compare a variety of data structures (1b, 2a, 2b)  
   ● Design efficient algorithms (2g, 6a)  
   ● Apply the knowledge gained in the class in order to solve real-world problems using different data structures and design techniques (1c, 1d, 2e)  
   ● Implement software solutions in C++ in the Linux OS environment (6a, 7g)  
   ● Use industry standard tools for software development including Git, Valgrind, and testing tools (1e, 6b, 6d)

7. **Brief list of topics to be covered**  
   ● Software development in C++  
   ● Introduction to the C++ Software Template Library (STL)  
   ● Advanced data structures and algorithms:  
     ○ Hashtables: Separate chaining, linear/quadratic probing  
     ○ Tree: BST, AVL, Red-Black, B+  
     ○ Heaps  
     ○ Sorting: Bubblestort, Insertion sort, quicksort, Heapsort, Mergesort  
     ○ Graphs: Dijkstra’s algorithm, Topological sort  
   ● Algorithm analysis with Big-O for both time and space complexity  
   ● Empirically comparing algorithms for space and time in different applications  
   ● Linux use, notably command line tools g++ and GNU Make  
   ● Git version control to manage software projects
• Valgrind, and software testing tools