

## CSC 330 Class Information and Syllabus

COURSE NUMBER: CSC 330  
COURSE TITLE: Advanced Data Structures  
CREDITS: 3:3  
PREREQUISITES: Grade of at least C (2.0) in CSC 230 and CSC 250

INSTRUCTOR: Name: Steve Tate  
Office: Petty 166  
Office Hours: Mon/Tues 1:30–3:30, or by appointment  
Phone: 336-256-1033  
E-mail: [srtate@uncg.edu](mailto:srtate@uncg.edu)

CLASS WEB SITE: <http://www.uncg.edu/cmp/faculty/srtate/330/>

CATALOG DESCRIPTION: Static and dynamic data structures emphasizing binary trees and graphs. Advanced programming techniques. Advanced sorting and searching algorithms. Hashing techniques. Performance analysis. Methods of developing large applications programs.

STUDENT LEARNING OUTCOMES: Upon successful completion of this course students will have demonstrated that they

1. understand inheritance and abstract classes;
2. can design divide-and-conquer algorithms using three steps and apply to merge sort, quick sort, dynamic programming, and backtracking;
3. understand tree representation and traversals;
4. understand graph representation and traversals; and
5. know associative containers, 2-3-4 trees, and red-black trees.

In addition to the above academic outcomes, students will gain practical experience working with a Unix-style system throughout the course.

**TEACHING METHODS AND ASSIGNMENTS FOR ACHIEVING LEARNING OUTCOMES:**

Class will meet twice a week for 75 minutes. Class time will include brief overviews of the material, but students are expected to have completed the textbook readings before class so that the majority of class time can be used for discussing the material, answering questions and clarifying topics from the book, and working through examples. If students do not keep up with readings, the instructor may give in-class quizzes or small reading-summary assignments as extra motivation.

**Assignments:** Assignments will be both written and programming. Programs must be written in C++, and must compile and work correctly on `cmpunix.uncg.edu` (students will be given login information and a brief system overview before the first assignment). The instructor will *not* make any adjustments to a student's code when grading, so if any submitted program does not compile the student will get a zero on the "correctness" portion of the grade (or 50 points off the overall grade), with no exceptions — more information on grading criteria will be distributed and discussed before the first assignment. Program source code will be turned in electronically, and students are expected to turn in a printout in class or to the instructor.

**Late Policy and Makeup Exams:** Assignments are due at the *beginning of class* on the due date, and may be turned in up to 7 calendar days late with a 25% late penalty. *No assignment will be accepted more than 7 calendar days after the original due date!* If the instructor adds assignments to ensure that students keep up with reading assignments, these will not be accepted late for any reason.

Exam/test dates are on the schedule on the following page — if there are any changes, they will be announced at least two weeks in advance if possible. A missed exam may be made up *only* if it was missed due to an extreme emergency and arrangements are made *before* the exam date. Exams (including the final) may not be taken early or late due to personal travel plans.

**EVALUATION AND GRADING:** Each student activity will contribute to the final grade in the class according to the following percentages.

|                                     |           |
|-------------------------------------|-----------|
| Assignments                         | 50 points |
| Mid-semester exams (15 points each) | 30 points |
| Final exam                          | 20 points |

**REQUIRED TEXTS/READING/REFERENCES:**

William Ford and William Topp. *Data Structures with C++ Using STL, Second Edition*, Prentice Hall, 2002. ISBN 0-13-085850-1.

## TOPICAL OUTLINE/CALENDAR:

| Date   | Topic   | Reading   |
|--------|---|-----------|
| Aug 24 | Class Introduction and Overview                               |           |
| Aug 26 | CSC 230 Review, Unix Overview, Programming Style              |           |
| Aug 31 | Trees: Definitions and Traversals                             | 10.1–10.4 |
| Sep 2  | Trees: Definitions and Traversals – cont'd                    |           |
| Sep 7  | <i>Labor day – no class</i>                                   |           |
| Sep 9  | Binary Search Trees and Tree Implementations                  | 10.5–10.7 |
| Sep 14 | Associate Containers: Sets and Maps                           | 11.1–11.3 |
| Sep 16 | Multisets and Implementations                                 | 11.4–11.5 |
| Sep 21 | Hashing   | 12.1–12.5 |
| Sep 23 | Hashing – cont'd  |           |
| Sep 28 | Exam 1  |           |
| Sep 30 | 2-3-4 Trees   | 12.6      |
| Oct 5  | Red-Black Trees   | 12.7      |
| Oct 7  | Object-Oriented Design: Inheritance                           | 13.1–13.2 |
| Oct 12 | <i>Fall break – no class</i>                                  |           |
| Oct 14 | Polymorphism, Virtual Functions, and Abstract Classes         | 13.6–13.7 |
| Oct 19 | Heaps and Priority Queues                                     | 14.1–14.3 |
| Oct 21 | Heaps and Priority Queues – cont'd                            |           |
| Oct 26 | Binary files  | 14.4      |
| Oct 28 | Divide-and-Conquer and Recursive Function Analysis            | 15.1      |
| Nov 2  | Divide-and-Conquer and Recursive Function Analysis – cont'd   |           |
| Nov 4  | Exam 2  |           |
| Nov 9  | Recursive Combinatorics                                       | 15.2      |
| Nov 11 | Dynamic Programming   | 15.3      |
| Nov 16 | Backtracking  | 15.4      |
| Nov 18 | Graphs: Terminology, Representations, and Traversals          | 16.1–16.5 |
| Nov 23 | Graphs: Terminology, Representations, and Traversals – cont'd |           |
| Nov 25 | <i>Thanksgiving – no class</i>                                |           |
| Nov 30 | Graphs: Minimization Algorithms – Shortest Paths              | 16.6      |
| Dec 2  | Graphs: Minimization Algorithms – Minimum Spanning Trees      |           |
| Dec 7  | Class Review  |           |
| Dec 14 | Final Exam (7:00–10:00PM)                                     |           |

**ACADEMIC INTEGRITY POLICY:** Students are required to sign the Academic Integrity Pledge on any work they do. The pledge is the statement “I have abided by the UNCG Academic Integrity Policy on this assignment.” For information on the UNCG Academic Integrity Policy, see <http://academicintegrity.uncg.edu/>.

Assignments in this class are for individual work, unless explicitly stated otherwise. General concepts and material covered in the class may be discussed with other students or in study groups, but specific assignments should not be discussed and any submitted work should be done entirely your own. The instructor uses a program comparison system that compares submissions and highlights programs that are too similar in order to detect cheating.

It is expected that the class textbook will be used as a reference, but if any other reference materials (including web sites) are used in preparing homework solutions they should be clearly cited. Any incidents of academic dishonesty will be handled strictly, resulting in either a zero on the assignment or an F in the class, depending on the severity of the incident, and incidents will be reported to the appropriate UNCG office.

**ATTENDANCE POLICY:** Attendance will not be taken in class, and is voluntary; however, all students are responsible for everything done or said in class (this can include changes in assignments, due dates, etc.). It is the student’s responsibility to obtain notes from another student if they miss class — the instructor will not provide notes.

#### **ADDITIONAL REQUIREMENTS:**

**Laptop/Cellphone Policy:** Laptops can be both a benefit and a distraction in a classroom. While many students benefit from taking notes using a laptop, or having access to outside class-related resources during class, other students cannot resist the temptation of checking e-mail, chatting, or even playing games during class time. This class has a strict “no non-class related use” rule for laptops — if you are found violating this policy, then your in-class laptop privileges will be taken away. Cellphones are a distraction for everyone, and should be turned off during class. If there is a special situation where you need to have your phone on for a particular day, please let the instructor know the situation before class.

**ADA STATEMENT:** UNCG seeks to comply fully with the Americans with Disabilities Act (ADA). Students requesting accommodations based on a disability must be registered with the Office of Disability Services located in 215 Elliott University Center: (336) 334-5440.

**UNIVERSITY CLOSINGS:** If university facilities are closed due to flu outbreak or other emergencies, it does not mean that classes are cancelled. In such an event, please check the class web page and Blackboard site for information about if and how the class will proceed.