1 Course Personnel

Instructor: Jason O’Kane (jokane@cse.sc.edu)
Teaching assistant: Shervin Ghasemlou (sherving@email.sc.edu)

2 Webpage

Information about the course, including scheduling information, assignments, and announcements, will be posted at this site:

https://cse.sc.edu/~jokane/teaching/350

3 Lectures

Tuesdays and Thursdays, 10:05–11:20am, Innova 1400 SWGN 2A14

4 Office Hours

<table>
<thead>
<tr>
<th>When?</th>
<th>Where?</th>
<th>Who?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mondays, 12:00–1:00</td>
<td>Innova 1205</td>
<td>Ghasemlou</td>
</tr>
<tr>
<td>Tuesdays, 3:00–4:30</td>
<td>Innova 2229</td>
<td>O’Kane</td>
</tr>
<tr>
<td>Wednesdays, 12:00–1:00</td>
<td>Innova 1205</td>
<td>Ghasemlou</td>
</tr>
<tr>
<td>Thursdays, 3:00–4:30</td>
<td>Innova 2229</td>
<td>O’Kane</td>
</tr>
</tbody>
</table>

You may also stop by my office at other times—I am often, but not always, around and available to help—or make an appointment using the site linked from the course webpage.

5 Description

Official description  Techniques for representing and processing information, including the use of lists, trees, and graphs; analysis of algorithms; sorting, searching, and hashing techniques.

Course outcomes  Algorithms and data structures form the backbone of computing. This course covers techniques for designing and analyzing algorithms for a variety of problems. After taking this course, you should be able to:

- Describe formal analysis measures.
- Describe the relevance of abstraction to problem solving.
- Analyze and use lists, trees, and graphs.
- Apply common algorithm design techniques such as brute force, divide-and-conquer, decrease-and-conquer, transform-and-conquer, dynamic programming, and the greedy technique.
- Analyze algorithms.
- Use appropriate data structures.

**Relationship to other courses** This course is different from many other computing courses, especially those at the lower level, because the primary content of the course is not focused on any specific programming language. (Several programming assignments will use C++, but only because of the need for uniformity, rather than because of any specific features of that language.) Instead, we will study algorithms that are useful across all programming languages. As a result, the things you will learn in this course should still be useful in, say, twenty years, when your favorite programming language is only a faded memory.

**6 Prerequisites**

*By course*  CSCE 240 (Advanced Programming Techniques); MATH 174 or 374 (Discrete Structures)

*By topic*  Introductory programming and data structures; Discrete mathematics; Use of simple data structures (queues, stacks, lists, trees)

**7 Textbook**

The required textbook is:


We will follow this book fairly closely, and homework problems will be assigned from it. However, the tests will be closed book.

**8 Evaluation and Grading**

Your learning in this course will be evaluated based on the following factors:

- An online **start-up quiz** quiz available at the course website. The quiz is due at 11:59pm on **August 27**. This quiz is worth **10 points**.

- **Homework** assignments will be given frequently. These assignments will be due at the start of the following class period. The purpose of these assignments is to give you a chance to practice the material covered in class, and to give an indication of the level of understanding that will be expected for the tests.
  - Submit the homework assignments on paper, at the start of class. Write your first name, last name, and the assignment number at the top of each sheet. The assignment number should match the assignment number on the course website. If you submit multiple pages, please staple them together.
If you consult external resources, other than the textbook, to complete the homework assignments, you must cite them explicitly. Even with a citation, you may not turn in answers written by someone else. Representing the work of others as your own is plagiarism; see policy below.

We will not invest class time returning homework assignments. If you would like to reclaim your homework papers after they are graded, may collect them from the teaching assistant.

The homework assignments will be worth **130 total points**.

- Four **programming projects** will be given through the semester. The purpose of these projects is to illustrate how to translate from the abstract algorithms we'll discuss in the class into working software. The projects must be done in C++ and will be graded using the department’s Linux platform. To expedite grading, each assignment will have very specific submission instructions, including precise formats for input and output. You will be expected to follow these instructions carefully. The first project, worth **10 points**, will be a very simple exercise to ensure that you understand the mechanics of how to complete the projects in the course. The remaining three projects will be worth **50 points each**.

- **In-class tests** will be given on October 2 and November 8. These will be closed-book exams with no electronic devices allowed. Each test will be worth **100 points**.

- A **final exam**, at 9am on December 13. The final will cover material from the entire course, with a focus on material introduced after the second in-class test. The final is worth **200 points**.

Thus, there are a total of **700 points** available to earn through the semester. The following table shows how final grades will be determined.

<table>
<thead>
<tr>
<th>Grade</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>630–700</td>
</tr>
<tr>
<td>B+</td>
<td>595–629</td>
</tr>
<tr>
<td>B</td>
<td>560–594</td>
</tr>
<tr>
<td>C+</td>
<td>525–559</td>
</tr>
<tr>
<td>C</td>
<td>490–524</td>
</tr>
<tr>
<td>D+</td>
<td>455–489</td>
</tr>
<tr>
<td>D</td>
<td>420–454</td>
</tr>
<tr>
<td>F</td>
<td>0–419</td>
</tr>
</tbody>
</table>

**Gradebook access** Grades will be posted on the CSE moodle server:

https://dropbox.cse.sc.edu

It is your responsibility to verify that grades are correctly recorded on this site.

**Corrections and regrades** My goal is to ensure that all of the grading for this course is fair and correct. If you believe there’s been a mistake in grading, please bring it to the instructor’s attention in office hours within one week after the scores are posted. Regrade requests after one week will be politely declined.
Important reminder  Keep in mind that I am grading your work, not you as a person.

Deviations from the grading policy  I assume that every student takes the class intending to succeed, and I share that goal. However, requests for grade increases that are inconsistent with the stated grading scale will be politely declined. Here is an incomplete list of hypothetical requests from students that are not sufficient for me to deviate from the stated grading scale:

- I need a GPA of at least ____ to get the internship I want.
- If my grade is less than ____, my parents will be disappointed in me.
- If my grade is less than ____, I won’t be able to graduate.
- I’ve never gotten a grade as low as ____ before.
- I have too many other responsibilities.
- The course is too hard.

9 Policies

Attendance  The instructor will not keep detailed records of class attendance. However, your instructor will make every effort to ensure that class attendance is worth your time. Missed tests due to unexcused absences will result in a score of zero. In the instructor’s experience, lack of class attendance has correlated strongly with poor grades.

Cheating  Academic dishonesty undermines the educational mission of the course and reflects disrespect to your classmates and to your instructor. Therefore, you are expected to practice the highest possible standards of academic integrity. The academic penalty for cheating is a failing grade for the course. This policy includes all forms of academic misrepresentation. Details on university’s academic integrity policies are available at

http://sc.edu/academicintegrity

Collaboration  Assignments should be done independently. It is permissible to discuss the problems at a high level with your classmates, but you should work out the details and compose the complete answers independently. Submission of identical or substantially identical work will be considered strong evidence that cheating has occurred.

Late assignments  Homework assignments cannot be accepted late, because we’ll discuss the solutions on the due date. Programming assignments submitted late will be assessed a penalty of 15% of the maximum score for each day or fraction of a day. Example: An assignment that would normally be scored 95/100, but submitted 2.5 days after the deadline, would earn 95 – ⌈2.5⌉ · 15 = 95 – 45 = 50 points.

Mobile devices  Please silence any mobile devices before coming to class. If your phone rings in class, I reserve the right to answer it for you and take a message. Likewise, if my phone rings during class time, I will allow a student to answer it.
Accommodations for disabilities  Any student with a documented disability should contact the Student Disability Resource Center to make arrangements for appropriate accommodations.

Policy changes  Changes to the syllabus at the instructor’s reasonable discretion, including changes to the evaluation and grading mechanisms, are possible but unlikely.