Data Structures and Algorithms (CS 361) Syllabus

Required Textbook:

Additional Resources (optional):
Additional materials are listed for those interested in more advanced approaches to common problems; however, the textbook is the main informational source for this course. The instructor’s lecture notes cover several topics not included in this textbook. These will answer specific questions regarding implementation and design patterns:

• More Effective C++ by Scott Meyers (ISBN: 020163371X) – Addison-Wesley Professional
• Effective STL by Scott Meyers (ISBN: 0201749629) – Addison-Wesley Professional
• Effective Modern C++ by Scott Meyers (ISBN: 1491903996) – O’Reilly Media
• C++ Coding Standards by Andrei Alexandrescu and Herb Sutter (ISBN: 0321113580) – Addison-Wesley Professional
• Exceptional C++ by Herb Sutter (ISBN: 0201615622) – Addison-Wesley Professional

Prerequisites:
CS 250 – Problem Solving and Programming or CS 333 – Problem Solving and Programming with C++
CS 252 – Intro to UNIX for Programmers
MATH 163 – Pre-Calculus II

Communications:
This online course will not have regularly scheduled lectures. For the most part, students will work at their own chosen times, subject to deadlines for assignments and exams. Blackboard announcements will list a limited number of optional network conferences, which will be recorded for the convenience of those unable to attend. Whereas this offers students scheduling flexibility, it also requires significant maturity and self-discipline to avoid falling behind on coursework. There are several options for course communications:

• Scheduled Net Meetings – a weekly Q&A session will be scheduled on most Mondays from 7-8 pm Eastern.
• Email – send specific questions regarding assignments or grade questions directly to the instructor.
• Office Hours – office hours will be held on Google Hangouts at the request of individual students. Requests should be made via email at least 24 hours in advance.

Note: This is a self-paced, self-taught, internet-based course. This means my role is less a lecturer and more a tutor/grade. Students are responsible for learning the material and asking questions using the various communication methods available to them.

Important: All email related to this course should have the subject-line prefix “CS 361:”. This flags your message in my inbox for faster attention. I will try to respond to all properly marked emails within 24 hours Monday - Friday, and within 48 hours on weekends and holidays.

Course Description and Organization:
This course explores data structures, algorithms for manipulating them, and the practical problems of implementing those structures in real programming languages and environments. Heavy emphasis is placed upon analysis of algorithms to characterize their worst- and average-case requirements for running time and memory.

CS 361 should expand the student’s toolbox of basic techniques for manipulating data at both the conceptual and concrete level. At the conceptual level, students will see a broad selection of standard practices and techniques used in program design. At the concrete level, students will begin a career-long practice of accumulating useful, reusable code units.

Assignments and Grading:
Each part contains several topics. Each topic is addressed in the textbook readings, lecture notes, and a variety of activities, including:

• Self-Assessments – ungraded online quizzes
• Labs – ungraded activities to practice techniques used in assignments
• Assignments and Quizzes – graded activities (most involve programming and analysis)
• Exams – administered after each part (first two mostly covering their respective parts with a cumulative final)

<table>
<thead>
<tr>
<th>Activity</th>
<th>Allocation</th>
<th>Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part I Exam</td>
<td>10%</td>
<td>≥ 94</td>
<td>A</td>
</tr>
<tr>
<td>Part II Exam</td>
<td>10%</td>
<td>90 – 94</td>
<td>A-</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20%</td>
<td>87 – 90</td>
<td>B+</td>
</tr>
<tr>
<td>Quizzes</td>
<td>25%</td>
<td>84 – 87</td>
<td>B</td>
</tr>
<tr>
<td>Assignments</td>
<td>35%</td>
<td>80 – 84</td>
<td>B-</td>
</tr>
<tr>
<td>Total</td>
<td>100%</td>
<td>77 – 80</td>
<td>C+</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74 – 77</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td></td>
<td>70 – 74</td>
<td>C-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>67 – 70</td>
<td>D+</td>
</tr>
<tr>
<td></td>
<td></td>
<td>64 – 67</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td></td>
<td>60 – 64</td>
<td>D-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt; 60</td>
<td>F</td>
</tr>
</tbody>
</table>
Software Requirements:
The compiler used for this course is the Free Software Foundation’s g++ (also known as gcc or GNU CC) version 4.8.4 or higher. Unless otherwise specified, it utilizes the C++11 standard. This is the compiler the instructor will use in evaluating and grading your assignments. If you have access to other compilers, you may use them; however, it is your responsibility to ensure they can be compiled by the official compiler.

You may wish to develop your programs on the most convenient compiler and then port it to the CS Department’s Linux computers for final testing. Please do not underestimate the amount of time involved in coping with subtle differences between compilers!

You can do all work using g++ on the CS Department Linux servers using SSH or X-Windows; however, you may obtain a g++ compiler for free from a variety of sources.

Computer Accounts:
Students will need 2 network accounts to participate in this class:
- An ODU ITS (Midas) account. This is the account associated with your @odu.edu email. It allows you to log into the course’s Blackboard site. All ODU students automatically receive this account, though you may need to activate it.
- An account on the ODU CS Department network. This is used to access CS Department resources. If you do not already have an account, set one up by going to https://www.cs.odu.edu and clicking “Account Creation” under the “Online Services” section.

Note: All students are responsible for ensuring they have working accounts prior to the first assignment!

Course Policies:
Assignments
- Follow the submission directions in each assignment’s requirements.
- Each assignment provides specific requirements that must be met. It is the student’s responsibility to ensure they meet the requirements on the submitted work.
- Grades for each assignment will be based on the rubric provided with the assignment.

Due Dates and Late Submissions
Students will have an average of 1 assignment due each week of the semester. Late submissions will not be accepted without prior arrangements. There are no make-up exams. Only university exceptions will be allowed (i.e., unusual situations or unforeseen circumstances outside the student’s control). For conflicts with foreseeable deadlines, prior arrangements must be made.

Suggestions for Success:
Students are highly encouraged to adopt successful behaviors:
- Write test cases before attempting to solve the problem. Determine how you will test your solution before you write your solution. Test-Driven Development (TDD) is an important skill to develop.
- Pay close attention to all necessary scenarios when designing your test cases. Failed test cases usually mean an unaccounted for edge case and should be addressed after the first assignment submission.
- Be systematic (i.e., 1 review the assignment, 2 review your testing procedures, 3 reevaluate your test cases).
- Be familiar with the UNIX command line, makefiles, and debugging tools.

Academic Honesty:
Everything turned in for grading in this course must be your own work. The instructor reserves the right to question a student orally and/or in writing and use his evaluation of the student’s understanding of the assignment and of the submitted solution as evidence of cheating. Violations of the Honor Code will be reported to the Office of Student Conduct and Academic Integrity. This includes students who contribute to violations by sharing their solutions.

This policy is not intended to prevent students from providing legitimate assistance to one another. Students are encouraged to collaborate to aid in the learning process. General assistance on the subject matter of the course is acceptable and encouraged. Specific discussions of solutions to any graded work are forbidden.

Honor Pledge:
I pledge to support the honor system of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagiarism. I am aware that as a member of the academic community, it is my responsibility to turn in all suspected violators of the honor system. I will report to Honor Council hearings if I am summoned.

By attending Old Dominion University, you have accepted the responsibility to abide by this code. This is an institutional policy, approved by the Board of Visitors.

Educational Accessibility:
Old Dominion University is committed to ensuring equal access to all qualified students with disabilities in accordance with the Americans with Disabilities Act (ADA). The Office of Educational Accessibility (OEA) is the campus office that works with students who have disabilities to provide and/or arrange reasonable accommodations.

- If you experience a disability which will impact your ability to access any aspect of the course, present me with an accommodation letter from OEA so that we can work together to ensure that appropriate accommodations are available to you.
- If you feel that you will experience barriers to your ability to learn and/or complete examinations in the course but do not have an accommodation letter, consider scheduling an appointment with OEA to determine if academic accommodations are necessary.

The Office of Educational Accessibility is located at 1021 Student Success Center, and their phone number is (757)683-4655. Additional information is available at the OEA website (http://www.odu.edu/educationalaccessibility/).