CSCE 2100 Syllabus

Fall, 2017

Instructor: Philip Sweany
Office: NTDP F262
Phone: 940-369-7427
Office Hours: Thursday from 11:15am .. 12:30pm and 2:00 .. 4:15pm.
Email: sweany@unt.edu

Textbook: (Available on line – for free !!!)
Foundation of Computer Science, Aho and Ullman

Course Description:
Computing Foundations I. Introduces students to formalisms used in computer science including (1) asymptotic behavior of algorithms, (2) combinatorics and probability, (3) trees and (5) graphs and (6) regular expressions

Course Outcomes:
Course outcomes are measurable achievements to be accomplished [by the student] by the completion of a course. Outcomes are evaluated in ABET's accreditation process.

1. Demonstrate a solid foundation in conceptual and formal models by describing loop structures in summation and/or product notation.
2. Use abstraction in the design and implementation of algorithms.
3. Design programming solutions to “simple” problems and implement those designs in C or C++.
4. Apply big-Oh notation to evaluate and compare algorithms and programs.
5. Use combinatorics and conditional probability in solving real-world problems.
6. Define the basic operations of sets, functions, relations, trees and graphs.
7. Demonstrate an introductory knowledge of formal languages by using regular expressions, deterministic finite automata and non-deterministic finite automata to describe patterns in strings.
Policies:

- All three exams, 2 midterms and a final, will be held in the university testing center (I hope). In each case you’ll have a window of at least 3 days in which to find a time that fits your schedule to take the exam.
- All of your programs (two of them) will be written in C or C++. For each programming assignment, the instructions will tell you whether to use C or C++. In addition, you should use a C or C++ that is compatible with that compiled on the “CSE” machines within the CSE department. If your program does not compile with the CSE gcc or g++ (whichever is specified for a particular assignment) and without any special flags, it will be counted as a program that has compile errors (which it does). EXCEPTION: You can, if you choose, submit a makefile with your program, in which case your program will be “compiled” using the makefile. It still needs to be written in C or C++ but it can use whatever compile flags you like, as long as it works on the CSE machines. But be warned. If your makefile doesn’t work properly with whatever code you submit, your program will be considered to have compiler error(s).
- You will NOT use the Standard Template Library in your programs this term in 2100. Failure to follow this “guideline” will result in a 0 for the program. The one exception will be that you CAN use the String class, which doesn’t belong in the STL, anyway!
- Most programs are due at 11:59pm on the due date.
- No late programs will be graded.
- All programs will be submitted to Blackboard.
- Attendance will not be taken in lecture. However, your attendance is strongly recommended to improve your opportunity to meet course outcomes. In addition, part of your grade (see grade breakdown below) will be based upon work done in class. Of course, if you are not in class when this work is done your grade will suffer.

Grading: The components of your grade will be weighted as follows:

- Recitation Quizzes, 10%
- In-class Exercises, 10%
- 2 Midterm exams, 20% each
- 1 final exam, 20%
- Two programs, 20% -- Note that you need to achieve and average of 60% on the two programs in order to “pass” the course (get an ‘A’, ‘B’, or ‘C’).

Often students assume that I grade on a 90/80/70/60 scale. That is NOT the case. I will guarantee that if your overall average is 90% or higher you’ll receive an ‘A’. If it is 80% or higher you’ll receive at least a “B”. If it is 70% or higher you’ll receive at least a “C”. Beyond that I give no guarantees.

Make-Up Policy:

There will be no make-up exams, recitations, or programs given in this class. However, for documented excused absences or emergencies on a day of an exam or a recitation, the exam and/or recitation grades will be replaced by an average of the other exams or recitation scores. There is one exception to this rule. Under NO circumstances will more than one exam or recitation score be
replaced by an average of the other scores. For a second (and subsequent) missed exam or lab, even if all are excused, students will receive a 0 for the missed work.

**Excused Absences**

Students are expected to schedule routine appointments and activities to not conflict with attending class. However, some absences cannot be prevented. In the event of a medical emergency or family death, students must request an excused absence as quickly as feasible following the emergency. Use common sense. Students must provide documentation to verify the emergency.

**Emergencies**

By definition, emergencies cannot be planned for. Your instructor attempts to make accommodations in these instances that allow for making up missed work and completion of the course in a timely manner. Among these emergencies are

- A death in your immediate family
- An accident or illness requiring immediate medical treatment and where a doctor has indicated attending class is impossible or inadvisable.
- Employees on call 24/7 must document that they were called during class time.

**Collaboration and Cheating:**

**Programming:**

Do **NOT** work with other students on shared program solutions. Do **NOT** get help with algorithms or coding from anyone other than Dr. Sweany, Mr. Helsing (teacher of the other section), the 2100 TA(s), or peer mentor(s). Do **NOT** use even partial program solutions from the internet unless those partial solutions are provided to you by Dr. Sweany as part of the assignment description. Failure to adhere to these strict standards will be cause for disciplinary action that could be as severe as expulsion from the university. **Those who are deemed to have cheated on program will receive a penalty that exceeds getting a 0 score for the program. Of course students who are deemed to have cheated have an appeal process of any penalty I invoke. If and when the time comes, that appeal process will be described to the student(s) in question.**

It IS permissible to obtain help from whoever you wish to fix **syntax errors**. But remember, for anything but syntax errors, getting programming assistance from any source other than Dr. Sweany, Mr. Helsing or the 2100 TAs or peer mentor(s) will be considered cheating and dealt with harshly.

**NOTE:** I’m have added a new caveat to my program grading policy this term, due to a “problem” that arose in Spring 2016. A student completed a programming assignment and placed the solution on GitHub. Another student accessed the GitHub repository, copied the solution and then made it available for other students to use. Thirteen students thus turned in a copy of the same solution and were penalized. One is no longer a UNT student. I subsequently learned that the department has had several cases where solutions placed on GitHub found their way into submissions by students who did NOT write the code. So, in future, students in my class who place their solutions on GitHub or any other unprotected site will be held equally responsible as any student who submits a copy of that program. If you wish to debate the use of GitHub or some other public repository, drop by my office during office hours and we’ll have a spirited conversation about it.
Recitations and “Lectures”

In stark contrast to programs, quizzes and exams I STRONGLY encourage students to work in groups both in the “lecture” classes and in recitation. I’ll explain my reasoning in class but let me say here that if you don’t take advantage of the cooperative learning available class and in recitation you’ll be missing out both on learning foundations in 2100 and in any future career you might pursue.

Exams:

And, of course you need to do your own work on quizzes and exams as well. Here there should be no ambiguity at all.

In case the above description, and in-class discussion of my views on appropriate and inappropriate collaboration does not answer all of your questions, please look at the university Student Rights and Responsibilities web page.

SPOT

The Student of Teaching (SPOT) survey is a requirement for all organized undergraduate classes at UNT. This short survey will be made available to you at the end of the semester, providing you a chance to comment on how this class is taught. I am very interested in the feedback I get from students, as I work to continually improve my teaching. I consider the SPOT to be an important part of your participation in this class.

ADA

UNT complies with all federal and state laws and regulations regarding discrimination, including the Americans with Disability Act of 1990 (ADA). If you have a disability and need a reasonable accommodation for equal access to education or services please contact the Office of Disability Accommodation.
## Tentative Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Reading</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/28</td>
<td>Computer Science as Abstraction</td>
<td>S1.1, pages 1 .. 9</td>
<td>“Assignment 0” 9/18/17</td>
</tr>
<tr>
<td>9/11</td>
<td>Iteration, Induction</td>
<td>S2.1 .. 2.3</td>
<td></td>
</tr>
<tr>
<td>9/18</td>
<td>Recursion</td>
<td>S2.5 .. 2.7</td>
<td></td>
</tr>
<tr>
<td>9/25</td>
<td>Run-Time, big-Oh</td>
<td>S3.1 .. 3.4</td>
<td></td>
</tr>
<tr>
<td>10/2</td>
<td>Run-Time big-Oh</td>
<td>S3.5 .. 3.8</td>
<td></td>
</tr>
<tr>
<td>10/9</td>
<td>More Big-Oh</td>
<td>S3.9 .. 3.12</td>
<td>Program 1 due 10/15 at 11:59pm</td>
</tr>
<tr>
<td>10/16</td>
<td>Combinatorics</td>
<td>S4.1 .. 4.4</td>
<td>Midterm 1</td>
</tr>
<tr>
<td>10/23</td>
<td>Combinatorics</td>
<td>S4.5 .. 4.8</td>
<td></td>
</tr>
<tr>
<td>10/30</td>
<td>Trees</td>
<td>S5.1 .. 5.4</td>
<td></td>
</tr>
<tr>
<td>11/6</td>
<td>Sets</td>
<td>S7.1 .. 7.3</td>
<td></td>
</tr>
<tr>
<td>11/13</td>
<td>Sets</td>
<td>S7.7, 7.10</td>
<td></td>
</tr>
<tr>
<td>11/20</td>
<td>Graphs</td>
<td>S9.1, 9.2, 9.6</td>
<td>Midterm 2</td>
</tr>
<tr>
<td>11/27</td>
<td>Automata</td>
<td>S10.1 .. 10.5</td>
<td>Program 2 due 11/27 at 11:59pm</td>
</tr>
<tr>
<td>12/4</td>
<td>Special Topics (or catch up)</td>
<td></td>
<td>Who knows?</td>
</tr>
<tr>
<td>12/8</td>
<td>Reading Day – no class – final exam will also be in Sage Hall, on one of 12/8, 12/11, 12/13 during hours listed in the Blackboard page for the class.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/9</td>
<td>Finals</td>
<td>Course Objectives</td>
<td>Final Exam</td>
</tr>
</tbody>
</table>

*Weeks are abbreviated as follows: 8/28/17 = August 28, 2017.*