CSCE 1040 Syllabus

Instructor: Philip Sweany
Office: NTDP F262
Phone: 940-369-7427
Office Hours: Tuesday, Thursday 2:00 – 3:30; Wednesday, 12:45 – 2:45
Email: sweany@cs.unt.edu

Textbook:
A Book On C (4th edition)
Al Kelley and Ira Pohl

Course Description:
CSCE 1040, the second course in the introductory sequence, focuses on more advanced C programming, designing and implementing larger software projects, introduction to dynamic data structures, and a brief introduction to C++ I/O and classes. The main focus is on developing students’ software development skills.

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

1. Write readable, efficient, and correct C programs that include programming structures such as assignment statements, selection statements, loops, arrays, pointers, both standard library and user-defined functions, dynamic memory allocation and deallocation, any subset of C’s rich set of operators and multiple header (.h) and code (.c) files.
2. Design and implement recursive algorithms in C.
3. Describe the concept of an abstract data type (ADT).
4. Use a combination of interactive and recursive design and implementation techniques to implement and use dynamically-allocated data structures in developing C applications.
5. Use a symbolic debugger to find and fix runtime and logical errors in C software.
6. Using a software process model, design and implement a significant software application in C. Significant software in this context means a software application with at least five files, ten functions and a makefile.
7. Implement, compile and run a small C++ program that includes a class definition and a main function to test the functionality of that class.
Policies:

- All Major programs are due at 11:59pm on the due date.
- No late programs will be graded
- All programs will be submitted to Moodle.
- Each student will be enrolled in a lab section and is expected to attend that lab each week.
- Except for the start of the term, attendance will not be taken in lecture. However, your attendance is strongly recommended to improve your opportunity to meet course outcomes. In addition, a portion of your grade will depend upon exercises completed in class.

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

- 1 midterm exam, 15%.
- final exam, 20%; (10% for written exam in the “lecture” final, and 10% for “lab practical” exam during the assigned final time for your lab section)
- Weekly labs, 25%; (Since we have two lecture sections and students from each lecture section will be enrolled in each lab section. It is important that you attend the lab in which you are enrolled. Do NOT expect that you can show up to a lab section other than the one to which you are assigned and be given “credit” for work completed in that lab for any week.)
- Four multi-week major programs assigned in lecture, 30%.
- Assignments given in the lecture class, whether they be quizzes or assignments to be turned in at the end of class, 10%.
- In order to pass the class a student MUST pass each of four different parts of the course grading, namely the
  - Lab portion of the course,
  - Major programs,
  - Lecture portion of the final exam and
  - Lab practical portion of the final exam.

Make-Up Policy:
There will be no make-up exams, labs, or programs given in this class. However, for documented excused absences or emergencies on a day of an exam or a lab the exam and/or lab grades will be replaced by an average of the other exam or lab scores. There is one exception to this rule. Under NO circumstances will more than one exam or lab 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 so as not to 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 that verifies an emergency arose.

**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 who are on call 24/7 fall in this category but must document that they were called during a scheduled class.

**Collaboration and Cheating:**

A major reason we have added a dedicated lab component to 1040 this year is that it provides an opportunity for some collaboration. Students (and even teachers?) generally can learn better and accomplish more working in collaboration with their peers. Because of that we have broken the weekly lab session into three parts, namely

1. A question and answer period to start the lab session. This gives students a chance to ask questions, nominally of the lab TA. But often the answers to students’ questions (during this time) will include discussion among all lab participants; students, TA, and (potentially) undergraduate tutors. (NOTE: of the 25 % of your grade associated with lab, one fifth of that (5 % of your total grade) will be awarded based upon your participation in this Q&A portion of the lab.)

2. A tutorial portion of the class will follow the question and answer session. In the tutorial (normally scheduled for 90 minutes) students will read and work through prepared material, learning and applying new programming techniques. Again, student teams are encouraged as long as the focus is on all members of the team learning the material and demonstrating that by completing the specified exercises. Of course the TA and (potentially) undergraduate tutors will be available for assistance. (NOTE: of the 25 % of your grade associated with lab, two fifths (10 % of your total grade) will be awarded based on participation in the tutorial portion of the lab.)

3. An “individual” programming assignment that each student is expected to complete in the remaining (1 hour) of lab. The goal of this portion of the lab is for students to practice and sharpen their skills as individual programmers. It is also the intent to prepare students for the lab practical (programming) portion of the final exam. The “default” for these individual lab programming assignments is that each student will work alone without use of any hard-copy or online materials EXCEPT the text book for the course. It
may well be that for some of the early assignments, collaboration will be allowed in which case your TA will let you know. But in general the collaboration rules for this lab component will be the same as for the major programs described next. (NOTE: of the 25% of your grade associated with lab, two fifths of that (10% of your total grade) will be awarded based on your participation in the individual programming portion of the lab.)

In contrast to the labs, on major programs assigned in lecture, you MUST work alone. Do NOT work with other students on shared program solutions. Do NOT get help with algorithms or coding from anyone other than Dr. Sweany or your 1040 lab TA. 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.

However, even for major programs it IS permissible to obtain help from whomever you wish to fix compilation errors. But remember, for anything but compiler errors, getting programming assistance from any source other than Dr. Sweany or your 1040 lab TA will be considered cheating and dealt with harshly.

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.

SETE
The Student Evaluation of Teaching Effectiveness (SETE) 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 SETE 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.
**Note:** In the schedule below, though each week (except the first) is shown to start on Monday, for purposes of the Lab only, each week will be considered to start on Friday. So, the lab associated with the 3rd week, (starting 9/16) will ACTUALLY start on Friday, 9/13 and continue from 9/16-9/19. Then the 4th “week’s” lab (Experiment) will start on 9/20 and continue through 9/26. The reason for starting each lab week on Friday is that the last Friday of the term is “reading day” and we can’t hold a lab on that day.

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Reading</th>
<th>Program</th>
<th>Lab</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/28/13</td>
<td>Introduction</td>
<td>Chapter 6 (as “review”)</td>
<td>Lab 1, OR</td>
<td></td>
</tr>
<tr>
<td>9/9/13</td>
<td>Arrays and Strings</td>
<td>Sect. 6.10-11</td>
<td>Lab 1</td>
<td></td>
</tr>
<tr>
<td>9/16/13</td>
<td>Recursion</td>
<td>Sect. 5.14-15</td>
<td>Lab 2 – I/O</td>
<td></td>
</tr>
<tr>
<td>9/23/13</td>
<td>Hash Table</td>
<td>ONLINE</td>
<td>Major Pgm 1; 9/27/13</td>
<td>Lab 3 – Recursion</td>
</tr>
<tr>
<td>9/30/13</td>
<td>Pointers, String Manipulation</td>
<td>Sect. 6.2-13</td>
<td>Lab 4 – Experiment</td>
<td></td>
</tr>
<tr>
<td>10/7/13</td>
<td>Bits, Bytes, Sets</td>
<td>Chapter 7</td>
<td>Lab 5 – Pointers</td>
<td></td>
</tr>
<tr>
<td>10/14/13</td>
<td>Structures</td>
<td>Chapter 9</td>
<td>Major Pgm 2; 10/18/13</td>
<td>Lab 6 – Recursion II</td>
</tr>
<tr>
<td>10/21/13</td>
<td>Midterm Exam</td>
<td></td>
<td>Lab 7 – Structures</td>
<td></td>
</tr>
<tr>
<td>10/28/13</td>
<td>Developing large programs</td>
<td>Sect. 5.8-13, 6.14, 8.11, 11.17</td>
<td>Lab 8 – Structures II</td>
<td></td>
</tr>
<tr>
<td>11/4/13</td>
<td>Lists</td>
<td>Sect 10.1-4</td>
<td>Major Pgm3; 11/8/13</td>
<td>Lab 9 – Gdb, makefile</td>
</tr>
<tr>
<td>11/11/13</td>
<td>Lists</td>
<td>Sect. 10.5-6</td>
<td>Lab 10 – Lists</td>
<td></td>
</tr>
<tr>
<td>11/18/13</td>
<td>Binary Trees</td>
<td>Sect. 10.8-9</td>
<td>Lab 11 – Stacks, Qs</td>
<td></td>
</tr>
<tr>
<td>11/25/13</td>
<td>G++</td>
<td>Chapter 13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12/2/13</td>
<td>G++ Classes</td>
<td>Chapter 13</td>
<td>Major Pgm 4; 12/5/13</td>
<td>Lab 12 – G++</td>
</tr>
<tr>
<td>12/9/13</td>
<td>Finals</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>