Syllabus

CS 2574: Introduction to Data Structures & Software Engineering
Second Summer Session, 1998

Index Number: 2300
Home page: http://ei.cs.vt.edu/~cs1704

Instructor: Craig A. Struble
Office: McBryde 521
Office Hours: 2:00–3:00, MTWH and by appointment
Email: cstruble@vt.edu
Phone: 231-5275

GTA: Mir Farooq Ali
Office: McBryde 116/118
Office Hours: To Be Determined
Email: mfall@csgrad.cs.vt.edu

Class Meets: Holden 110, 11:00–12:15, Monday through Friday
Final Exam: 8:00–10:00, Saturday, August 8, 1998

Prerequisites

- Programming in C, either CS 1044 or CS 1344;
  CS majors must have successfully completed the prerequisite with a C or better. There will be NO exceptions.
- Corequisite: Tools class, either CS 1206 or CS 2304 (UNIX);
  Any CS major successfully completing the corequisite before taking CS 2574 must have a C or better in order to take CS 2574. There will be NO exceptions.

Description

This class introduces students to the skills needed to design and implement medium to large sized programming projects. The projects are implemented using modules, basic data structures, and software engineering techniques.
Textbook

Required


The course notes are available online in PDF format and the complete course note pack can be purchased from A-1 Copies in the University Mall.

References

Additional references are given in the course note pack. When possible, the books are on reserve in the library.

Grading

Grading for this class will be on a 1000 point scale. The breakdown of the points is as follows:

<table>
<thead>
<tr>
<th>Component</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Project 1</td>
<td>250</td>
</tr>
<tr>
<td>Programming Project 2</td>
<td>300</td>
</tr>
<tr>
<td>Midterm</td>
<td>200</td>
</tr>
<tr>
<td>Final Exam</td>
<td>250</td>
</tr>
</tbody>
</table>

There is no formal curve given for any single project or test. The final grades will be determined based on your relative performance in the class. I do guarantee that anyone getting 900 points or higher receives at least an A-, 800 points at least a B-, 700 points at least a C-, etc.

Programming Assignments

There are two sizeable programming projects. It is possible that the second project will build on the first, so it is important to keep up with the material and have a working first program. The program grades consist of 60% for functionality and 40% for documentation and coding style. Programs MUST adhere to the *Elements of Programming Style*\(^1\) to receive full credit for documentation.

In addition to comments in code, each assignment is accompanied by two structure charts presenting the design of the program. The first structure chart is due ONE WEEK BEFORE the final version of the program is due, and represents the initial design of the programming project. The second structure chart is due with the program and represents the final design of the program. It is expected that the two structure charts will differ, so do not worry about getting it right the first time.

Programs must be submitted in class on the day they are due. Programs may be submitted up to 1 day late (not counting weekends), for a 20% penalty. Programs later than 1 day will not be accepted. No extensions will be given for hard drive crashes, network outages, or scheduled lab closings. Make sure you keep frequent backups of your work.

Further information regarding assignment submissions is available on the home page. This information must also be followed to receive full credit on the program.

\(^1\)http://ei.cs.vt.edu/~cs1704/Standards/Standards.html
Programming Environment

Students may use any programming environment they wish to develop their programs. However, all programs must compile and execute using the Microsoft Visual C++ installed in the McBryde 116/118 lab. Submitted programs must execute correctly on the lab machines under Windows NT. Programs that do not execute correctly on the lab machines will have points deducted. The GTA will not go to dorm rooms, homes, or other labs for program demonstrations. It is also not acceptable to bring your own system (e.g., laptop or PC) to demonstrate your program.

Collaborative Work

Students may pair up to work on the programming assignments. The complete policy is available at the URL http://ei.cs.vt.edu/~cs1704/collaborate.html.

Tests

There will be one midterm and one final examination. These tests may be of any format. The tests cover material presented in class as well as from readings in the textbooks. No make up tests will be given unless the instructor is notified at least 24 hours in advance that you will be missing a test and a valid excuse must be given to warrant a make up test.

Honor Code

The VPI & SU honor code is in effect for this course. In addition, the CS Department’s “Policy on koofers, old programs, cheating, and computer use”\(^2\) also applies to this course. Additional University, Departmental, and legal policies may also apply to this course; see the home page for details.

Note

If any student needs special accommodations because of a disability, please contact the instructor during the first week of classes.

\(^2\)http://ei.cs.vt.edu/~dept/koofers.html
# Course Schedule

<table>
<thead>
<tr>
<th>Dates</th>
<th>Topics</th>
<th>Readings</th>
<th>Special</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/29</td>
<td>Course administration, syllabus, intro software engineering</td>
<td>Standish Chap. 1, notes</td>
<td></td>
</tr>
<tr>
<td>6/30–7/1</td>
<td>intro software engineering, structure charts</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7/2–7/7</td>
<td>Misc. C topics: Standard I/O, Separate Compilation</td>
<td>Standish pg. 95–96, notes</td>
<td></td>
</tr>
<tr>
<td>7/8</td>
<td>Pointers</td>
<td>Standish Chap. 2</td>
<td></td>
</tr>
<tr>
<td>7/9–7/13</td>
<td>Linked lists, double and circular linked lists</td>
<td></td>
<td>7/10 Program 1 Structure Chart due</td>
</tr>
<tr>
<td>7/14–7/15</td>
<td>Recursion</td>
<td>Standish Chap. 3</td>
<td></td>
</tr>
<tr>
<td>7/16–7/17</td>
<td>Abstract Data Types</td>
<td>Standish pg. 111–135</td>
<td>7/17 Program 1 due</td>
</tr>
<tr>
<td>7/20–7/21</td>
<td>ADTs: Stacks, Queues</td>
<td>Standish Chap. 7 &amp; 8</td>
<td></td>
</tr>
<tr>
<td>7/22</td>
<td>MIDTERM</td>
<td></td>
<td>MIDTERM</td>
</tr>
<tr>
<td>7/23–7/24</td>
<td>ADTs: Drop-out stacks, dequeues</td>
<td>Standish Chap. 7 &amp; 8</td>
<td></td>
</tr>
<tr>
<td>7/27</td>
<td>Algorithm Analysis</td>
<td>Standish Chap. 6</td>
<td></td>
</tr>
<tr>
<td>8/3</td>
<td>Software lifecycle process</td>
<td>Standish Chap. 5 &amp; 16</td>
<td></td>
</tr>
<tr>
<td>8/4–8/5</td>
<td>Table Driven Software</td>
<td>Standish Chap. 14, notes</td>
<td>8/5 Program 2 due</td>
</tr>
<tr>
<td>8/6</td>
<td>Testing</td>
<td>Standish Chap. 5</td>
<td></td>
</tr>
<tr>
<td>8/8</td>
<td>8:00–10:00 Final Exam</td>
<td></td>
<td>Final Exam</td>
</tr>
</tbody>
</table>

Note, this schedule is subject to change at the instructor’s discretion.