Fall 9-1-2000

CS 441.01: Theory and Practice Advanced Programming

Joel Henry
*University of Montana - Missoula, joel.henry@umontana.edu*

Follow this and additional works at: [https://scholarworks.umt.edu/syllabi](https://scholarworks.umt.edu/syllabi)

Let us know how access to this document benefits you.

**Recommended Citation**


[https://scholarworks.umt.edu/syllabi/4897](https://scholarworks.umt.edu/syllabi/4897)

This Syllabus is brought to you for free and open access by the Course Syllabi at ScholarWorks at University of Montana. It has been accepted for inclusion in Syllabi by an authorized administrator of ScholarWorks at University of Montana. For more information, please contact scholarworks@mso.umt.edu.
Course Syllabus
CS 441 Theory and Practice Advanced Programming

PREREQUISITE: CS 335, CS 344, CS 346, CS 365 and Math 325 or consent of instructor.

COURSE OBJECTIVES:
This is your capstone sequence in the Computer Science Department and therefore this course will extend and deepen your understanding of topics previously covered. Specifically the objectives of this course are to gain an understanding of regular and context free languages, and language processing. You will examine and implement advanced data structures, algorithms, and programming techniques. You will understand design tradeoffs, software testing, maintenance and documentation issues, and program execution and memory requirements. Issues specific to graphical user interfaces and real-time systems will also be covered. All these issues will serve as the basis for a major project in CS 442.

INSTRUCTOR:
Name: Joel Henry
Office: Social Sciences 411
Office Hours: MWF 9-10; Tues & Thurs 1-2 If the door is open I am available
E-mail address: henryj@cs.umt.edu
Phone: 243-2218

TOPICS:
I. Introductory programming issues.
II. Languages and language processing.
III. Implementation issues.
IV. Compilation and linking issues.
V. Software design issues.
VI. Software testing, maintenance and documentation issues.
VII. Real-time systems, data compression, multithreading/signaling.
VIII. Preparing for a major project.

TEXTS:
REQUIRED: Compiler Construction, Louden
RECOMMENDED: Code Complete, McConnell ($28 from Amazon)

COURSE CONTENT:
I have changed this course from the material previously taught by Dr. Ford. These changes are to make this course the best course I can possibly provide to you. I will be using material from the Louden text but supplementing course materials heavily from other texts, journal articles, and personal experience. Course materials from other sources will be given to you or you can obtain them from the WWW or from reserve at the library. Please note: You may use Java or C++ in this course, and I have extensive professional computing experience which will influence this course heavily. You will not find this information in any textbook so COME TO CLASS!
COURSE DELIVERABLES:
- Tests (3 @ 20% each): 60%
- Assignments (7): 40%

GRADING:
Grading scale: 59.5 or lower F, 59.6-69.5 D, 69.6-79.5 C, 79.6-89.5 B, 89.6 - 100 A

LATE POLICY:
Hand in materials:
- 1 day (24 hours) - 10%
- 2 days (48 hours) - 30%
- 3 days (72 hours) - 50%
- 4 days or more - No thanks, I don’t want it.

Presentations:
- No late presentations.

CHEATING:
Plagiarism will be handled harshly, as per the Student Conduct Code. You may fail the assignment or the course. MY ADVICE: Take an F rather than cheat.

COURSE INFORMATION:
This course has changed to take advantage of the strengths of the instructor and to provide course materials that meet the vision of the instructor. In short, I asked myself, “What materials would have been most useful me in a capstone course prior to graduation?” This course answers that question.

You have never had me for a course and so have no stories from other students to rely on. Here is a cliff notes version of what you might hear if you were to talk to former students of mine. I am tough but fair. I have little tolerance for trash handed to me disguised as coursework. “The dog ate my homework” stories do not fly. If you have a legitimate problem or obstacle to submitting assignments or taking tests, I will go the extra mile to help you. I am informal in dress and presentation. Do not let this belie the fact that I take this job and this material very seriously. You will work hard but I will be working hard along side you to provide challenging, interesting, and relevant materials. I make mistakes and have no problem admitting them. However, I am bright and experienced and will not tolerate any disrespect. I will not disrespect you.

I firmly believe you can succeed in this course, can complete every assignment, and can master every test question. I believe you will meet my high expectations. If you are working hard at this course, I will do all I can to insure you get the grade you wish. If you attempting to do as little as possible to succeed, or just plain being lazy, you will find me indifferent to your success.

Come to class on time, just as you would if working for a prospective employer. If I am talking you are not talking. This is also a behavior I get upset with very quickly. If you want to chat, don’t come to class. If you are talking when I am, you will get to leave.
**Computer Science 441 - Syllabus**

**Tentative Schedule:**

<table>
<thead>
<tr>
<th>Week</th>
<th>Tuesday</th>
<th>Thursday</th>
<th>Material</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sept. 5 – Introduction, Syllabus, Course Information, first assignment</td>
<td>Sept. 7 – Compilers</td>
<td>Louden Chapter 1</td>
<td>Design portion of assignment 1 due Sept. 7</td>
</tr>
<tr>
<td>2</td>
<td>Sept. 12 – Scanning tokens</td>
<td>Sept. 14 – Regular expressions and Lex</td>
<td>Louden Chapter 2</td>
<td>Coding portion of assignment 1 due Sept. 12</td>
</tr>
<tr>
<td>3</td>
<td>Sept. 19 – Grammars</td>
<td>Sept. 21 – Parsing</td>
<td>Louden Chapter 3</td>
<td>Assignment 2 due Sept. 21</td>
</tr>
<tr>
<td>4</td>
<td>Sept. 26 – Top-down Parsing</td>
<td>Sept. 28 – Recursive descent parser</td>
<td>Louden Chapter 4</td>
<td>Assignment 3 due Sept. 28</td>
</tr>
<tr>
<td>5</td>
<td>Oct. 3 – Return homework and Review for Test 1</td>
<td>Oct. 5 – Test 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Oct. 10 – Review Test 1, Software design</td>
<td>Oct. 12 – Design analysis and measurement</td>
<td>Handouts and class notes</td>
<td>Assignment 4 due Oct. 19</td>
</tr>
<tr>
<td>8</td>
<td>Oct. 24 – Software Compilation and Linking</td>
<td>Oct. 26 – Software Deployment &amp; Upgrades</td>
<td>Handouts and class notes</td>
<td>Assignment 5 due Nov. 2</td>
</tr>
<tr>
<td>9</td>
<td>Oct. 31 – Software testing</td>
<td>Nov. 2 – Software testing</td>
<td>Handouts and class notes</td>
<td>Assignment 5 due Nov. 2</td>
</tr>
<tr>
<td>10</td>
<td>Nov. 7 – No class, Election Day Holiday</td>
<td>Nov. 9 – Software maintenance</td>
<td>Handouts and class notes</td>
<td>Assignment 6 due Nov. 14</td>
</tr>
<tr>
<td>11</td>
<td>Nov. 14 – Preparing for a team software project</td>
<td>Nov. 16 – Implementing a team project</td>
<td>Handouts and class notes</td>
<td>Assignment 6 due Nov. 14</td>
</tr>
<tr>
<td>12</td>
<td>Nov. 21 – Test 2</td>
<td>Nov. 23 – Thanksgiving Holiday</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Nov. 28 – Real-time software</td>
<td>Nov. 30 – Real-time software</td>
<td>Handouts and class notes</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Dec. 5 – Data Compression</td>
<td>Dec. 7 - Data Compression</td>
<td>Handouts and class notes</td>
<td>Assignment 7 due Dec. 12</td>
</tr>
<tr>
<td>15</td>
<td>Dec. 12 – Multithreaded applications</td>
<td>Dec. 14 – Multithreaded applications</td>
<td>Handouts and class notes</td>
<td>Assignment 7 due Dec. 12</td>
</tr>
<tr>
<td>16</td>
<td>Wednesday, December 20 3:20-5:20</td>
<td>Final Exam</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Large Scale C++ Design, Lakos, Design Patterns, Gamma et. al., Code Complete, McConnell
2. Large Scale C++ Design, Lakos, Code Complete, McConnell
3. Large Scale C++ Design, Lakos, Dynamics of Software Development, McCarthy
4. Software Testing, Myers
5. Software Engineering, Pressman
6. Software Engineering, Pressman, Dynamics of Software Development, McCarthy
7. Software Engineering, Pressman, Software Engineering, Sommerville
8. Lecture notes
9. Lecture notes