Winter 2007

CS 241-01: Computer Science II

Eric Maston
Wright State University - Main Campus

Follow this and additional works at: http://corescholarlibraries.wright.edu/cecs_syllabi
Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation
http://corescholarlibraries.wright.edu/cecs_syllabi/743

This Syllabus is brought to you for free and open access by the College of Engineering and Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact corescholar@wwwlibraries.wright.edu.
Computer Science II
Winter Quarter 2007
Wright State University

January 2, 2007

Course Description

This course is the second in the Introduction to Computer Science (24X) series. It focuses on object oriented concepts and an introduction to data structures.

Goals

There are several goals to accomplish in CS 241:
1. Master object oriented programming
2. Continue learning skills with IDE such as Visual C++.
3. Learn how to develop more complex software programs.
4. Have some fun!

Class Details

Lecturer: Eric Matson
Office: 336 Russ Engineering Center
Phone: 937-775-5108
Office Hours: Monday/Wednesday 4:00 - 5:45 at Russ 336 or by appt.
Email: eric.matson@wright.edu
Web: http://agents.cs.wright.edu/index.html, then select teaching link
Class: Tuesday/Thursday 6:05 - 7:20 FH 218
IDE: Microsoft C++ .NET
Prerequisites

For this class the official prerequisite is CS 240. Please let me know after the first lecture if you do not meet this prerequisite. We can talk about your preparation if it differs from the requirement. This section of CS 241 will be taught using the C++ programming language. Please see me immediately if you took CS 241 using Java.

Grading

Homework 30% 4 Programming Assignments  
Lab 20% 8 Lab Assignments  
Midterm Exams 30% 2  
Final Exam 20% 1

The base scale is: A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: 0-59. This is the highest requirement that will be used. The scales may be lowered or revised if necessary.

Schedule

(always subject to changes) Always have readings scheduled for that day complete prior to the class meeting

<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Topic</th>
<th>Reading</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jan 2</td>
<td>Introduction</td>
<td>Chapter 6</td>
</tr>
<tr>
<td>2</td>
<td>Jan 4</td>
<td>Review Arrays and Vectors</td>
<td>Chapter 9</td>
</tr>
<tr>
<td>3</td>
<td>Jan 9</td>
<td>Pointers</td>
<td>Chapter 10</td>
</tr>
<tr>
<td>4</td>
<td>Jan 11</td>
<td>Pointers</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Jan 16</td>
<td>Overloading, Review</td>
<td>Chapter 17</td>
</tr>
<tr>
<td>6</td>
<td>Jan 18</td>
<td>EXAM 1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Jan 23</td>
<td>Overloading, Name Scope Management</td>
<td>Chapter 20</td>
</tr>
<tr>
<td>8</td>
<td>Jan 25</td>
<td>Name Scope Management</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Jan 30</td>
<td>Recursion</td>
<td>Chapter 14</td>
</tr>
<tr>
<td>10</td>
<td>Feb  1</td>
<td>Recursion</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Feb  6</td>
<td>Sorting and Searching</td>
<td>Chapter 15</td>
</tr>
<tr>
<td>12</td>
<td>Feb  8</td>
<td>Sorting and Searching</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Feb 13</td>
<td>Dynamic Memory Allocation</td>
<td>Chapter 16</td>
</tr>
<tr>
<td>14</td>
<td>Feb 15</td>
<td>List</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Feb 20</td>
<td>List, Review</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Feb 22</td>
<td>EXAM 2</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Feb 27</td>
<td>Inheritance</td>
<td>Chapter 11</td>
</tr>
<tr>
<td>18</td>
<td>March 1</td>
<td>Inheritance</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>March 6</td>
<td>Polymorphism</td>
<td>Chapter 21</td>
</tr>
<tr>
<td>20</td>
<td>March 8</td>
<td>Polymorphism, Review</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>March 15</td>
<td>Final Exam</td>
<td>8:00 - 10:00 pm</td>
</tr>
</tbody>
</table>
Policies and Notes

- **Attendance**: Attendance is not required, nor will it be taken after the first couple of lectures. If you are not a regular attendee, it will be your responsibility to seek out what material was covered in the lecture and learn it. Most of my exam questions will be taken directly from ideas covered during the lecture, so it greatly helps if you attend!

- I will utilize WebCT to post updates to the course, sample code, projects, announcements, schedule, etc. Get in the habit of checking it regularly.

- The prerequisites of the course are basic understanding of high-level development in C++ and object oriented concepts. If you are not confident in your skills or do not have the required prerequisites, then visit with me and I can evaluate how to catch your skills up to the appropriate level and develop a plan to do so.

- Always make back ups of all of your work. Never have just one copy of anything!

- If you are going to miss an exam, for any reason, discuss it with me in advance. If it is an emergency situation, please notify me as soon as possible.

- You can reach me a number of ways. Email is normally the best as I check it about 18 hours a day normally. You can also reach me by phone during the day at 775-5108. If you need human contact either stop in during my office hours, make an appointment, or just come by my office. If I am in and not on a deadline to get something else completed, I will normally try to help as much as possible.

- There are technologies we will use in this class that you may not already know, such as file transfer, command line, text editors, file systems, etc. We will cover some of these technologies as we go.

- The key to learning in this class will be spending time working through the problems. Don't wait until 2 hours before something is due to try to learn the concept and then write the program. This normally ends in a disaster! Stay up with the readings and try to work through some of the examples in the book. I will post what I call, 10 minute programs which are exercises that you can work through to learn key concepts. And yes, they are programs you can write and execute in 10 minutes (unless you are a really slow typist, like me. In that case, they become 20 minute programs.)

**Academic Misconduct**

In this class, the only way to truly learn the concepts to is do the work yourself. I encourage working with other people on the course concepts. When you begin to write the program, complete and submit your own work.

Work that has obviously been copied or in the more extreme case, when the original authors name has not even been changed, both parties will receive a 0 grade for that assignment. Both parties will also be turned over to the Office of Judicial Affairs.