Fall 2004

CS 242: Introduction to Computer Science III

Praveen Kakumanu

Wright State University - Main Campus

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation

https://corescholar.libraries.wright.edu/cecs_syllabi/130

This Syllabus is brought to you for free and open access by the College of Engineering & 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 library-corescholar@wright.edu.
CS 242: Introduction to Computer Science III  
Fall 2004

Instructor: Praveen Kakumanu
Office: 358 Russ Engineering Center
Office Hours: T TH, 3:00 – 4:00 PM and by appointment
Voice: 775 – 5121
Email: kpraveen@cs.wright.edu
URL: http://www.cs.wright.edu/~kpraveen

Course Description
This is the final course in the three course sequence "Introduction to Computer Science" offered by the Computer Science department, WSU. It focuses on building a number of abstract data types such as stacks, queues, trees and tables. We continue to study the C++ object-oriented concepts such as Inheritance, polymorphism and template handling. We also start learning to analyze the complexity of algorithms in this course.

Note: For all CS 242 students, concurrent registration into CS 242 lab is a must.

Textbook

Language
Microsoft Visual C++ 6.0 (also available in WSU Dunbar Library).

Grading
The course grade will be the weighted sum of four grades. Grading will be straight scale (90-100 A, 80-89.9 B, 70-79.9 C, 60-69.9 D, below 60 F). These numeric thresholds may be lowered due to clustering, but will not be raised.

• Programming Projects
There will be four programming assignments to be done individually and handed in by the due date mentioned in the class. No late submissions are accepted. Programs must be written well in a modular fashion with proper indentation, style, and documentation. Programs will be graded based on correctness, efficiency and documentation.

• Laboratory Exercises
There is a lab section for this course, in which the students familiarize themselves with the concepts taught in the class or sometimes trying new concepts. Labs are handled by Graduate Teaching Assistants who will guide and check the student assignments. There will be eight laboratory assignments and each will include the following:
  • A prelab exercise due at the beginning of next lab section
  • An inlab assignment to be completed during the lab session
- Examinations
There will be one midterm and one final exam. The midterm will be held during the 5th week of the quarter. Missed
exams, if any, can be made up only in case of emergency or work conflicts and require proper supporting
documentation. The final exam is scheduled during the final week of classes and all students are required to take it
as per the announced schedule.

<table>
<thead>
<tr>
<th>Work load</th>
<th>Weight (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Programming Projects (@ 9%)</td>
<td>36</td>
</tr>
<tr>
<td>8 Laboratory Exercises (@ 3%)</td>
<td>24</td>
</tr>
<tr>
<td>1 Midterm (@20%)</td>
<td>20</td>
</tr>
<tr>
<td>1 Final Exam</td>
<td>20</td>
</tr>
</tbody>
</table>

Academic dishonesty
Students are encouraged to share ideas by discussing with others. However, all the work you submit should be of your
own. Submitting the code of others is regarded as cheating. All the students who are involved in such an activity will
receive a grade of F. Also read and understand the WSU policy for academic honesty and integrity
http://www.wright.edu/students/judicial/stu_integrity.html

Additional Information
Information regarding the course readings, assignments, labs and exams will be posted on the course web page.
Students are expected to check the web page on a regular basis for any updates. The instructor reserves the right to
modify any of the course policies, schedule and due dates.

Tentative Schedule

<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Readings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Software Engineering and Design</strong></td>
<td>Ch. 1, 2, 5</td>
</tr>
<tr>
<td></td>
<td>Problem Solving and Recursion</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><strong>Inheritance &amp; Virtual Functions</strong></td>
<td>Ch. 11.10 – 11.13, 15 (Gaddis)</td>
</tr>
<tr>
<td></td>
<td><strong>ADT Development Methodology, STL List Class</strong></td>
<td>Ch. 3, 4</td>
</tr>
<tr>
<td>2-3</td>
<td><strong>C++ Templates and Exceptions</strong></td>
<td>Ch. 16 (Gaddis)</td>
</tr>
<tr>
<td>4</td>
<td><strong>The Stack ADT, Stack Applications</strong></td>
<td>Ch. 6</td>
</tr>
<tr>
<td>5</td>
<td><strong>The Queue ADT, Queue Applications</strong></td>
<td>Ch. 7</td>
</tr>
<tr>
<td>6</td>
<td><strong>Searching and Sorting Algorithms</strong></td>
<td>Ch. 9</td>
</tr>
<tr>
<td>7-8</td>
<td><strong>Computational Complexity</strong></td>
<td></td>
</tr>
<tr>
<td>9-10</td>
<td><strong>Binary Search Trees</strong></td>
<td>Ch. 10, 11</td>
</tr>
<tr>
<td></td>
<td><strong>Priority Queues &amp; Heaps</strong></td>
<td></td>
</tr>
<tr>
<td>Finals Week</td>
<td>Final Exam: Thursday, November 18th, 8:00 – 10:00 P.M.</td>
<td>Everything!</td>
</tr>
</tbody>
</table>