Instructor
Dr. Deborah Hwang
Home page: http://cs215.ueharlax.ac.uk/~hwang

Course Home Page
Announcements regarding handouts and assignments will be made in class. Assignments will be available only at the course home page (http://cs215.ueharlax.ac.uk/~hwang/f07-courses/cs215.html). It is your responsibility to consult the course home page on a regular basis.

Catalog Data
Project and problem-solving course emphasizing the use of classes for encapsulation of abstract data types and abstract data structures. Topics include classes, templates, dynamic allocation, searching and sorting, recursion, and exception handling.

Objectives
To continue the study of problem solving techniques used in programming software solutions including abstract data types, templates, exception handling, dynamic allocation, recursion, sorting and searching algorithms, and basic algorithm analysis. To continue the study of the C++ language features that support these techniques. Exposure to the UNIX operating system and development tools.

Prerequisites: CS 210

Required Textbook

Daily and Weekly Requirements
Assigned daily reading assignments. Weekly homework assignments. Homework assignments may include short programming problems.

Programming Projects
There will be 7-9 programming projects of 1-2 weeks in duration each. See handout A C++ Programming Guideline for CS 215 for appropriate code format used in this course. It is similar, but not the same, as the style used in the textbook.

Programming projects will be graded using the following criteria with the weights as shown.

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>65%</td>
<td>Correct results, including command line arguments and file I/O</td>
</tr>
<tr>
<td>10%</td>
<td>Error checking, including proper use of exceptions</td>
</tr>
<tr>
<td>25%</td>
<td>Style, observed coding guidelines, originality</td>
</tr>
</tbody>
</table>

In addition, there will be an up to 3-point penalty if an appropriate Makefile is not submitted when required.

Programming projects should be submitted both on-line by email and in hard-copy printout as explained in the handout Submission Instructions for CS 215.
Exams and Evaluation
There will be two in-class written exams and a comprehensive final written exam. In addition, there is an evening lab programming practical exam that will be scheduled around midterm. The purpose of the practical exam is to demonstrate mastery in using the C++ programming language and the UNIX environment. Therefore, it is necessary to score a minimum of 60% on the practical exam to pass the course (grade of C- or better). Students who fail to do so and are otherwise passing the course will be given a second opportunity to pass the practical exam at the end of the term with a 10% penalty. Final grades will be based on the following weighted distribution:

- 20% Comprehensive final exam
- 20% Two in-class exams (10% each)
- 15% Lab programming practical exam
- 5% Homework (weighted as indicated in assignment)
- 40% Programming projects (weighted as indicated in assignment)

Final grades are based on the final weighted percentage with some adjustments depending on class distribution. Historically, the A/B line falls around 88% +/- 2% with subsequent grade levels every 10%.

Attendance Policy
The Harlaxton attendance policy will be applied to this course. Students are permitted absences equal to the number of times that a course meets per week. For this course, this is three lecture periods. Absences in excess of this number will result in a reduction in the semester grade by one level for each additional absence. Of course, you are encouraged to attend every class. Hopefully, you will learn much more by attending class AND reading the text than by skipping class and relying only on the text.

Students are responsible for all material covered in class. If you do miss a class, find out what was covered from another student.

Honor Code
All students are expected to adhere to the University's Honor Code regarding receiving and giving assistance. The following specific guidelines are in force for this course.

- **Homework exercises** (including programming homework exercises) are for you to gain experience and practice. You may collaborate with your classmates, but each student should submit a solution in his/her own words that reflect his/her understanding of the solution. This includes the programming exercises, which should be the result of your own typing. Ultimately you will be required to demonstrate your proficiency of the material on exams. Therefore, it is highly recommended that you attempt all homework problems on your own before finding a solution from another source.

- **Programming projects are to be your own work unless otherwise noted.** Discussing the meaning and general solution techniques of an assignment with other students is permitted. For example, discussing “How is this assignment similar or different from problems presented in the text or in lecture?” is acceptable.

Asking another person for assistance on **specific** items in your own analysis and design or code is also permitted, but you may not observe another person’s solution or code in its entirety for the purposes of studying or copying it, with or without that student's permission.
For example, asking, "What does this compiler error mean?" or "Do I have the correct class syntax here?" is acceptable. Whereas asking "Can I see how you coded your stack?" is not acceptable.

In particular, since UNIX systems tend to be open by default, it is absolutely forbidden to "rummage" around the Harlaxton server file system looking at anyone else's work even if they have not set the file permissions to prevent such observation. (For those that would rather not rely on the integrity of others, it is suggested that all work for this class be put into a subdirectory that has its permissions set to owner only.)

Giving or receiving unauthorized aid on a programming project will result in a 0 for the project on the first offense. Any subsequent violations will result in an F for the course and possibly formal disciplinary action.

- Exams, of course, are to be solely your own work. Giving or receiving any type of unauthorized aid on any exam will result in a final grade of F and possibly formal disciplinary action.

If there is any doubt as to whether assistance is acceptable, consult the instructor.

**Tentative Course Schedule**

<table>
<thead>
<tr>
<th>Week of</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wed</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>08/27</td>
<td>Introduction UNIX, g++, make</td>
<td>C string tutorial argc, argv</td>
<td>File streams</td>
<td>No class</td>
<td></td>
</tr>
<tr>
<td>09/03</td>
<td>Chapter 1.1-1.5 ADTs &amp; Classes</td>
<td>Chapter 1.7-1.7 C++ strings</td>
<td>Chapter 2.1-2.3 Exception handling String streams</td>
<td>No class</td>
<td></td>
</tr>
<tr>
<td>09/10</td>
<td>Chapter 2.4 Overloading</td>
<td>Chapter 3.1-3.2 Selection sort Search</td>
<td>Chapter 3.3-3.4 Algorithm analysis</td>
<td>No class</td>
<td></td>
</tr>
<tr>
<td>09/17</td>
<td>Chapter 3.6 Recursion</td>
<td>Chapter 3.7 Recursion</td>
<td>Chapter 3.5, 4.1-4.3 Templates, vectors</td>
<td>Chapter 4.3-4.4 Using vectors Insertion sort</td>
<td></td>
</tr>
<tr>
<td>09/24</td>
<td>Exam 1 Review</td>
<td>Exam 1 Material to 9/24</td>
<td>No class</td>
<td>No class</td>
<td></td>
</tr>
<tr>
<td>10/01</td>
<td>Chapter 5.1-5.2 Pointers dynamic arrays</td>
<td>Multi-dimensional dynamic arrays</td>
<td>Chapter 5.3-5.4 Dynamic classes</td>
<td>No class</td>
<td></td>
</tr>
<tr>
<td>10/08</td>
<td>Chapter 5.5-5.6 Dynamic classes</td>
<td>Practical Exam 2-HR TBA</td>
<td>Debugging, gdb</td>
<td>No class</td>
<td></td>
</tr>
<tr>
<td>10/15</td>
<td>Chapter 6.1-6.2 List ADT, Iterators</td>
<td>Chapter 6.3-6.4 List operations</td>
<td>Chapter 7.1-7.5 Stack ADT</td>
<td>No class</td>
<td></td>
</tr>
</tbody>
</table>

08/26/07 3 of 4
<table>
<thead>
<tr>
<th>Week of</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wed</th>
<th>Thursday</th>
<th>Friday</th>
</tr>
</thead>
</table>
| 10/22   | Chapter 7.4-7.5  
Chapter 8.1-8.3  
Stacks, Queue ADT | Chapter 8.4-8.5  
Queues | Chapter 9.1-9.3  
Linked Lists | No class |        |
| 10/29   | Chapter 9.4  
Using linked lists | Chapter 9.5-9.6  
Doubly-linked lists | Exam 2 Review | No class |        |
| 11/05   | **Exam 2**  
New material to  
10/30 | Chapter 10.1-10.3  
Binary trees | No class | No class |        |
| 11/12   | Chapter 10.3-10.4  
Traversing binary trees | Chapter 10.5  
Binary search trees  
BST ADT | Chapter 10.6-10.7  
Using BSTs  
BST class | Chapter 14.1-14.2  
Heaps, heapsort |        |
| 11/19   | Chapter 8.6, 14.3  
Priority Queue ADT | Chapter 15.1  
Quicksort | Chapter 15.1  
Quicksort | No class |        |
| 11/26   | Chapter 15.1  
Mergesort | Chapter 15.1  
Comparing sorting algorithms | Last day of class  
Course review | No class |        |

Note that there is class on Friday, September 21 and Friday, November 16. These class periods make up for missed classes on Thursday, September 27 (Ireland trip) and Thursday, November 8 (Paris trip).

Final exam is on Tuesday, December 4, 9:30am-11:30am.