Computer Science II — CSci 1200 — Spring 2007
Syllabus

Instructor:	Prof. Chuck Stewart 302 Materials Research Center (MRC), x6731 `stewart@cs.rpi.edu`, `stewart@rpi.edu`
Lecture:	Tue, Fri 2:00-3:20, Sage 3303
Office hours:	Tue / Fri after class, Wed, Thu 1:15-2:45
Course website:	`http://www.cs.rpi.edu/academics/courses/spring06/cs2`
Textbooks:	*Data Structures with C++ Using STL*, 2nd edition Ford and Topp, Prentice Hall *Accelerated C++* Koenig and Moo, Addison-Wesley

These books are **recommended only**. The course will follow primarily the order of the Ford and Topp book, which is available at the bookstore. Some of the lectures are based on material in Koenig and Moo. Students should have available a C++ reference book, such as those by Malik or Stroustrop, **and** use on-line reference material.

Lab Sections

Assignment of lab supervisors to lab sections will be announced via the course web site. Get to know your lab TA, and get to know some of the other students in your lab sections. Your TA will get to know you. Your TA is your first point of contact for this course. You may attend the office hours of any TA or the instructor, especially for help with the course material (instructor) and homework assignments (TAs).

Course Emphasis and Goals

This is a course in elementary data structures and their use in programming. This includes both class design and features of the C++ programming language. Much of our discussion will be built around the design and use of the C++ standard library (STL). By using the standard library, students will be able to write reasonably sophisticated programs quickly.

Prerequisite

Computer Science I or the equivalent. This includes some assumed background on computational problem solving and programming as well as the

following specific knowledge of C++: types, variables, arithmetic, assignment statements, i/o streams, logic, conditionals, if-then-else statements, while and for loops, functions, parameter passing, arrays and vectors. We will quickly review this material in the first two lectures and the first lab. Students who find this review difficult to follow should immediately undertake remedial work.

C++ vs. Java

Some students may enter this course having started with Java instead of C++. This should not be a major problem, and reasonably proficient Java programmers should stay in this course instead of taking CS I. The lecture and lab materials will be augmented occasionally with comparisonqs between some of the properties of the two languages as an aid to the transition.

Requirements and Grading

Semester requirements will include a combination of labs, homeworks, and exams. The weights in determining the semester average are as follows:

Labs:	12%
Homeworks:	33%
Tests:	35%
Final:	20%

Letter grades will be computed from the semester average. Averages will be rounded to the nearest integer and assigned letter grades as follows:

Avg	**Grade**
≥ 93	A
90-92	A-
87-89	B+
83-86	B
80-82	B-
77-79	C+
73-76	C
70-72	C-
67-69	D+
60-66	D
≤ 59	F

There will be 12 labs, one each Wednesday of the semester except February 21 (Winter Break week), March 28 (GM week) and May 2nd (the last

day of classes). Labs will be graded on a scale of 0-3, depending on the amount and quality of work completed. Labs will be designed so that students who work diligently can earn all 3 points. Students **must** attend their assigned lab sections unless prior permission has been given. Lab instructions and a subset of the lab problems will be posted on the course website on the Tuesday morning before lab. Additional problems will be distributed at the start of the lab.

Homework assigments will be made available via the course web site on Thursday mornings. Generally, they will be due the following Thursday, 7 days later, by 11:59:59pm. Some assignments will have a two-week duration. For the most part, homeworks will be programming problems. Submission will be done electronically. Instructions will be given with the first assignment.

Three tests will be given during the semester, with 5% of the semester average assigned for the worst test grade (for each individual student) and 15% for each of the other two. The cumulative final, given during finals week, will be worth 20%. This assigns 55% of the semester average for tests. In addition, **students must have an overall average of at least a 55% on the tests in order to pass the course!** This is 30.25 points of the 55 percentage points available on the tests.

Homework Late Policy

Homework assignments must be submitted electronically. Each student will be given **three days** (whole or partial) of grace for late homework assignments. These grace days should be used carefully, and **no more than two** may be used for any one assignment. Once the late days have been exhausted, late assignments **will not be accepted** without a written excuse from the Dean of Students' office.

As an example, if student BG submits his/her 1st assigment 26 hours late, BG will have used two late days and have only one day left. If BG then submits another assignment 5 hours late, BG will have used his/her last late day. If BG then submits a 3rd assignment 1 minute late, it will not be accepted. **Students should use their late days carefully, saving them for the latter part of the semester or, better yet, not using them at all.**

Academic Integrity

Copying, communicating or using disallowed materials during an exam is cheating, of course. Students caught cheating on an exam will receive an F in the course and will be reported to the Dean of Students office. Students are allowed to assist each other in labs, but must write their own lab solutions.

Academic integrity on programming assignments is a complicated issue. It is addressed in a separate statement handed out during the first lecture.

Refer to the Rensselaer Handbook for further discussion of academic dishonesty.

Schedule

The schedule of coverage and important dates for the semester are shown on the next page. Pointers to reading material will be provided with the lecture notes.

Lecture	Date	Coverage
1	1/16	Intro and background
2	1/19	Background continued, recursion
3	1/23	C++ Classes, part 1
4	1/26	C++ Classes, part 2
5	1/30	Pointers and arrays
6	2/02	Dynamic memory
7	2/06	Implementation of `std::string`
8	2/09	String and char operations
	2/13	**Test 1**
9	2/16	Lists
10	2/23	Iterators; examples
11	2/27	Linked lists, part 1
	3/02	**no class**
12	3/13	Linked lists, part 2
13	3/16	Associative containers, part 1
14	3/20	Associative containers, part 2
	3/23	**Test 2**
15	3/27	Problem solving and program design
16	3/30	Trees, part 1
17	4/03	Trees, part 2
18	4/06	Advanced recursion techniques
19	4/10	Hash tables, part 1
20	4/13	Hash tables, part 2
21	4/17	Priority queues, part 1
	4/20	**Test 3**
22	4/24	Priority queues, part 2
23	4/27	Inheritance and polymorphism
24	5/01	Inheritance and polymorphism

A number of topics, including stacks and queues as well as operators and friend classes will be covered in lab only and therefore are not listed here.