

CSCI-1200 Computer Science II — Spring 2006

Instructor:	Prof. Barb Cutler 309A Materials Research Center (MRC), x3274 cutler@cs.rpi.edu
Lecture:	Tuesdays & Fridays 10-11:20am, DCC 318
Office hours:	Tuesdays & Fridays after class
Teaching Assistants:	see webpage for contact info
Lab Sections:	Wednesdays, see webpage for times & locations
TA Office hours:	see webpage for times & locations
Course website:	http://www.cs.rpi.edu/academics/courses/spring06/cs2/
Required book:	<i>Accelerated C++</i> , Koenig and Moo, Addison-Wesley
Optional book:	<i>C++ Programming</i> , D.S. Malik, Thomson Learning (2nd edition)

The course will primarily follow the first book, which is quite dense. The second book covers some of the same material, especially the basics of C++, but much more comprehensively, with many examples and a slower pace. This second book should be purchased if you are not comfortable with the basics of C++ or if you need extra examples to learn new material. There are also two very good websites you should find helpful:

<http://www.sgi.com/tech/stl/>

is the STL (standard template library, or now just standard library) programmer's guide at SGI. It is a good resource for questions about vectors, lists, maps, etc.

<http://www.parashift.com/c++-faq-lite/>

is an organized summary of the most important FAQ's that have appeared on the C++ newsgroups. Both beginning and advanced students can benefit from it. Feel free to search for and share other web resources.

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

Early in the semester we will start programming using the standard C++ library. By using the standard library, students will be able to write reasonably sophisticated programs quickly. This will encourage the development of important problem solving skills. Later in the semester we will study how the container classes and functions of the standard library are implemented. In doing so, we will introduce many of the low-level tools of the C++ language. Thus, we are deferring many of the gritty but important language details until students have the programming maturity to handle them.

Prerequisites

Computer Science I or the equivalent. Assumed knowledge of C++ includes types, variables, arithmetic, assignment statements, i/o streams, logic, conditionals, if-then-else statements, while and for loops, functions, parameter passing, and arrays. We will review this material in the first two lectures and the first lab.

C++ vs. Java

Some students may enter this course having started with Java instead of C++. This shouldn't be a major problem, and if you are a reasonably proficient Java programmer, you should stay in this course instead of taking CS I. The lecture and lab materials are augmented with comparisons between some of the properties of the two languages as an aid to the transition.

Warning

This course is substantially more difficult than CS I and moves at a much more rapid pace. Students should not get behind any point in the semester, but especially not early. Students should work practice problems and study examples posted on the course web site. Working with other students and working with tutors and TAs are both encouraged, but students need to be certain they understand the material and can do problems on their own.

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:	15%
Homeworks:	30%
Tests:	35%
Final:	20%

There will be 14 labs, essentially one each week. Labs will be graded on a scale of 1-3, based 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 afternoon after lecture. Additional problems will be distributed at the start of the lab. **Note that students should take the Koenig and Moo textbook and recent lecture notes to lab.**

Homework assignments will be made available via the course web site on Thursday mornings. Generally, they will be due the following Thursday, 7 days later, by 11:59pm. Some assignments, especially later in the semester, will have a two-week duration. For the most part, homeworks will be programming problems. Submission will be done electronically. See the Homework Guidelines handout for more details.

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%. In addition, **students must have an overall passing average on the tests in order to pass the course!**

Homework Late Policy

Assignments will be due Thursday nights at 11:59pm. Submission will be electronic and submission times will be judged from the timestamp given by the web server; that is, our clock, not yours. If it is one minute late, it is a day late, so we suggest that you don't actually wait until 11:59 to send your homework, just to be on the safe side.

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 assignment 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

Here is the tentative schedule of lectures for the class. Please see the webpage for the updated schedule as the term progresses.

Lecture	Date	Coverage	Reading
1	1/17	Introduction and Background	Ch 0
2	1/20	Background II	
3	1/24	Strings	Ch 1, 2
4	1/27	Vectors	Ch 3
5	1/31	C++ Classes I	Sec 4.2-4.4, Ch 9
6	2/3	C++ Classes II	Sec 4.2-4.4, Ch 9
	2/7	Test 1	
7	2/10	Lists and Iterators	Sec 5.1-5.5
8	2/14	Iterators II	Sec 5.1-5.5
9	2/17	Recursion I	
10	2/24	Recursion II	
11	2/28	Problem Solving I	
12	3/3	String and Char Ops	Sec 5.6-5.9
	3/7	Test 2	
13	3/10	Associative containers I	Ch 7
14	3/21	Associative containers II	Ch 7
15	3/24	Problem solving II	
16	3/28	Pointers and Arrays	Sec 10.1-10.3
17	3/31	Dynamic Memory	Sec 10.4-10.7
18	4/4	Operators and Friends	
19	4/7	Vector Implementation	
	4/11	Test 3	
20	4/14	Linked Lists I	
21	4/18	Linked Lists II	
22	4/21	Trees I	
23	4/25	Trees II	
24	4/28	Problem Solving III	
25	5/2	Additional Topics and Review	
	TBA	Final Exam	

Homework Guidelines

This handout contains guidelines on due dates, submission and grading policy that will be in force throughout the semester. Please read it carefully and keep it accessible!

Due Date and Time

Assignments will be due Thursday nights at 11:59pm. Submission will be electronic and submission times will be judged from the timestamp given by the web server; that is, our clock, not yours. If it is one minute late, it is a day late, so we suggest that you don't actually wait until 11:59 to send your homework, just to be on the safe side.

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 assignment 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.**

Electronic Submission

Follow the instructions VERY carefully! To submit your homework, first make sure that the file you are submitting satisfies all of the requirements found in the homework description. Then, go to the homework subpage of the course website. The direct link to this page is

<http://www.cs.rpi.edu/academics/courses/spring06/cs2/hw.html>

You will need your RPI login/password to access this page. Using the radio buttons select the assignment that you are going to submit. **Double-check to make sure that you selected the correct one; otherwise, your assignment will be misplaced by the system.** Use the Browse button to select which file you want to submit and press the Send File button **only once**. If you have more than one file, please zip them up first and send the zip file. You do not need to worry about identifying your submission with any kind of special naming scheme (other than the file names you have used), and you don't need to inform your TA — all of that is taken care of automatically. You will either get a message confirming your submission or an error message advising you of a submission requirement you failed to adhere to. If, after submitting, you find a mistake in your homework, you can fix it and re-submit using the same procedure as before. We will **only grade the most recent submission**. You don't have to do anything special for resubmissions.

Compilers

You may use any C++ compiler you wish for this course, but your submitted code must run under either the latest Microsoft Visual Studio compiler or under gcc 3.x (3.3 is standard on Mac laptops). In the comments in your code, please indicate which compiler you used to test your program. When you submit your program, use the radio buttons to select the compiler you used for the homework.

Grading Guidelines

Here are some general comments about program grading. Points for programming assignments are generally divided into:

- Error-free compilation,
- Program structure, and
- Correct program output.

The program structure grade is an evaluation of the quality of your code. This includes the following:

- Your code should have a clear, logical organization. Functions should be written as appropriate to break up the program.
- Variable names should be intuitive and meaningful. For example, if you need a variable to represent an employee's salary, use the name `salary`, not `x`.
- Use comments to describe functions and significant sections of code. These don't have to be long and involved. Keep them short and clear.
- Use a consistent, logical method of indentation to make your program easy to read. For example, code inside a loop or if-else condition must be indented all at the same level (until reaching the inside of a nested loop or nested if-else construct).
- Don't type lines of text that are too wide to fit the screen. Break statements cleanly across multiple lines as needed.
- Avoid the use of global variables. Global constants are generally fine.

Use the example code given in class, in lab, and in the textbooks as guidelines. **Programs that are unclear and do not compile will earn little or no credit.**

Program correctness will be determined by evaluating your program on a series of test data sets. We will make available some but not all of this data, along with sample output, prior to the assignment due date. This will help you judge the quality of your program, but will not guarantee correct results on all data. You will need to ensure this on your own.

Notes and Warnings

- If you seek help on a programming assignment from a TA or from the instructor, you must be prepared with a list of problems and questions. Students who are not prepared will be turned away.
- **Do not place your files in a publicly accessible area.** The importance of this can not be stressed enough. If your code is stolen and submitted, even with significant alterations, you are likely to be caught and accused of academic dishonesty. See the statement on academic integrity.
- After you have submitted your assignment solution, do not modify the only copy of your code. If you do wish to make changes, perhaps to improve it, then make a copy before modifying your code.

Academic Integrity for Programming Assignments

Important: It will be assumed that you have read the following statement thoroughly. If you have any questions, contact the instructor or the TAs immediately.

Academic integrity is a difficult issue for programming assignments. Students naturally want to work together, and it is clear they learn a great deal by doing so. Getting help is often the best way to interpret error messages and find bugs, even for experienced programmers. In response to this, the following rules will be in force for programming assignments:

- Students are allowed to work together in designing algorithms, in interpreting error messages, and in discussing strategies for finding bugs, but NOT in writing code.
- Students may not share code, may not copy code, and may not discuss code in detail (line-by-line or loop-by-loop) while it is being written or afterwards. This extends up to two days after the submission deadline.
- Similarly, students may not receive detailed help on their code from individuals outside the course. This restriction includes tutors.
- Students may not show their code to other students as a means of helping them. Sometimes good students who feel sorry for struggling students are tempted to provide them with “just a peek” at their code. Such “peeks” often turn into extensive copying, despite prior claims of good intentions.
- Students may not leave their code (either electronic versions or printed copies) in publicly accessible areas. Students may not share computers in any way when there is an assignment pending.

We use an automatic code comparison tool to help spot assignments that have been submitted in violation of these rules. The tool takes all assignments from all sections and compares them, highlighting regions of the code that are similar. Code submitted by students who followed the rules produces less than 10% overlap. Code submitted by students who broke the rules produces anywhere from about 30% to 100% overlap.

We (the instructor and the TAs) check flagged pairs of assignments very carefully ourselves, and make our own judgment about which students violated the rules of academic integrity on programming assignments. When we believe an incident of academic dishonesty has occurred, we contact the students involved.

Students caught cheating on programming assignments will be punished. The standard punishment for the first offense is a 0 on the assignment and a 5 percentage point penalty on the semester average. Students whose violations are more flagrant will receive a higher penalty. For example, a student who outright steals another student’s code will receive an F in the course immediately. Students caught a second time will receive an immediate F, regardless of circumstances. Each incident will be reported to the Dean of Students’ office.