CSCI 1321 (Principles of Algorithm Design II), Spring 2001
Syllabus

1 Course description

This course is the second course for computer science majors, following the guidelines established by the Association for Computing Machinery. This course also partially satisfies the requirements for *Understanding the World Through Science* of the common curriculum. The course content will include defining data types including singly-linked lists, doubly-linked lists, stacks, queues, and trees; recursion; use of libraries; pointers; dynamic memory; type-independent programming; and program implementation strategies.

The objectives of this course include, but are not limited to, the following:

- Learning fundamental problem-solving methodology.
- Implementing algorithms using a programming language.
- Dealing with complex systems.
- Development and analysis of algorithms.
- Introduction to the basic topics in data structures.

2 Basic information

Class meeting times and location:

MWF 11:30am – 12:20pm, Halsell 228

Prerequisites:

CSCI 1320, or consent of instructor

Instructor:

Dr. Berna Massingill

Office: Halsell 201L

Office phone: (210) 999-8138

Web page: `http://www.cs.trinity.edu/~bmassing`

E-mail: `bmassing@cs.trinity.edu`

Office hours: MW 12:30pm – 3:30pm, TR 2:00pm – 4:00pm, and by appointment

3 Course materials

Textbook:

Michael Main and Walter Savitch. *Data Structures and Other Objects Using C++*. Addison Wesley, second edition, 2001.

Web page:

Most course-related information (this syllabus, homework and reading assignments, etc.) will be made available via the World Wide Web. The home page for the course is not only a starting point for Web-accessible course material but will also be used for course-related announcements. Please plan to check it frequently. You can find it linked from my home page (`http://www.cs.trinity.edu/~bmassing`) or directly at `http://www.cs.trinity.edu/~bmassing/CS1321_2001spring/`.

Other references:

- T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGraw Hill, sixth edition, 1992. Emphasizes algorithms, not programming.
- Cygnus. "The ISO/ANSI C++ Draft Standard". `http://www.cygnus.com/misc/wp/`. If you want to know about some obscure C++ rule and have lots of hours to understand the terminology.
- Daniel P. Friedman and Matthias Felleisen. *The Little LISPer*. MIT Press, trade edition, 1987. A good book on recursion.
- Daniel P. Friedman and Matthias Felleisen. *The Little Schemer*. MIT Press, fourth edition, 1995. A good book on recursion.
- Nicolai M. Josuttis. *The C++ Standard Library: A Tutorial and Reference*. Addison Wesley, 1999. A good STL reference book.
- Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Prentice Hall, second edition, 1989. *The* reference for C, written by its creators.
- Andrew Koenig and Barbara Moo. *Ruminations on C++*. Addison Wesley, 1997. A great but advanced book on C++ programming.
- Stanley B. Lippman. *Essential C++*. Addison Wesley, 2000. A very short book covering C++ essentials.
- Stanley B. Lippman and José Lajoie. *C++ Primer*. Addison Wesley, 1998. A lengthy book introducing C++ language features.
- P.J. Plauger, Alexander A. Stepanov, and Meng Lee. *Standard Template Library: A Definitive Approach to C++ Programming*. Prentice Hall, 1996. Presumably authoritative, since one of the authors wrote the STL.
- Bjarne Stroustrup. *The C++ Programming Language*. Addison Wesley, 1997. The creator of the C++ programming language introduces its features.

4 Course requirements

Grading:

Grades in this course will be determined by the results of three major exams and several homework assignments. Each midsemester exam will be worth 100 points, the final exam will

be worth 200 points, and together the homework assignments will be worth approximately 400 points, with the weight of individual assignments determined by their length and difficulty. Numeric grades will be calculated as a simple percentage, by dividing points earned (on homework assignments and exams) by points possible. These numeric grades will then be converted to letter grades based on a curve, but in no case will the resulting letter grades be worse than students would receive based on the following scheme:

Numeric grade	Letter grade
90 – 100	A
80 – 89	B
70 – 79	C
60 – 69	D
0 – 59	F

Exams:

Exams are comprehensive but will emphasize the most recent material. They are scheduled as follows. Please plan accordingly.

- Exam 1: February 16, in class. *Changed to:* February 21, in class.
- Exam 2: March 30, in class. *Changed to:* April 4, in class.
- Final exam: May 8, 8:30am.

Homework assignments:

Several homework assignments will be required for successful completion of this course. Detailed requirements, including due dates and times, will be provided as part of each assignment; normally you will have about a week per assignment. Most homeworks will be laboratory problems, which will be coded in a suitable programming language. You are encouraged to use the department's network of Unix machines, but unless otherwise specified for individual assignments, you may use any other system that provides a suitable environment.

Attendance:

Regular class attendance is strongly encouraged.

Late and missed work:

Exams can be made up only in cases of documented conflict with a university-sponsored activity or documented medical emergency.

Homework will normally be accepted up to one class period late, *but no more*, at a penalty of 10 percent off per working day. This penalty may be waived or additional time allowed *at the instructor's discretion* in cases of illness or conflict with a university-sponsored activity.

If you have unusual circumstances (as we all sometimes do), please discuss these with the instructor in advance.

Collaboration and academic integrity:

Unless otherwise specified, all work submitted for a grade (homework assignments and exams) must represent the student's own individual effort. Discussion of homework assignments among students is encouraged, but not to the point where detailed answers are being written

collectively. Answers that are identical beyond coincidence are in violation of Trinity's Academic Integrity Policy and will result in disciplinary action, including, but not limited to, a failing grade on that assignment for all parties involved. You are responsible for the security of your work, both electronic and hard copy.