# Computing & Software Systems 343:
# Data Structures and Algorithms
# Winter 2006

In this course, you will be introduced to the bulk of the basic abstract data types, algorithms, and computational models used by computer professionals. By the end of this quarter, you will be a confident C++ programmer and will be comfortable with the basics of object-oriented design and programming. You will understand how to analyze a problem and design a solution, recognizing when existing techniques and software are reusable. You will understand the tradeoffs among memory, running time, and implementation time associated with different data structures and algorithms. Topics include: trees, tables and priority queues, graphs, grammars, data abstraction, object-oriented design, OO programming, computational complexity and algorithm analysis.

**Instructor** Michael Stiber `<stiber@u.washington.edu>`, room UW1-341, phone 352-5280, office hours after class or by appointment.

**Course Web** http://courses.washington.edu/css343/stiber.

**Course Blog** http://css343.blogspot.com/.

**Classes** Tuesdays and Thursdays, 3:30–5:35, UW1-010 (though we will likely be meeting in the Center for Integrated Teaching, Learning, and Scholarship, UW1-302).

**Textbooks** Mark Allen Weiss, *Data Structures and Problem Solving Using C++*, Second Edition, Addison Wesley Longman, 2000.

Kenneth H. Rosen, *Discrete Mathematics and Its Applications*, Fifth Edition, McGraw Hill, 2003.

**Suggested Reading** Bruce Eckel, *Thinking in C++*, Second Edition, vols. 1 & 2, Prentice Hall. A good “from scratch” introduction to the language. Both volumes are available on-line at http://www.bruceeckel.com/.

Bjarne Stroustrup, *The C++ Programming Language*, Third Edition, Addison-Wesley, 1997. The canonical reference. Make sure you get the third edition.

Harley Hahn, *Harley Hahn’s Student Guide to UNIX*, 2nd edition, McGraw-Hill, 1996. If you’re unfamiliar with Unix or Linuxand want to learn more, get a book like this.

Herbert Schildt, *STL Programming from the Ground Up*,Osborne/McGraw-Hill, 1999. A very good introduction to the Standard Template Library, with lots of examples. However, it is not a reference; for example, it doesn’t provide complete lists of methods for each class.

Nicolai M. Josuttis, *The C++ Standard Library*,Addison-Wesley, 1999. Much more of a complete reference than the Schildt book. Includes some examples, but is not intended as a tutorial.

**Grading** Your course average is computed as: 30% project contributions + 30% class contributions + 20% midterm + 20% final

I don’t grade on a curve. I use my judgment to determine what averages correspond to an ‘A’, ‘B’, etc. for the quarter, after the fact. Some quarters assignments, etc. turn out harder, and so the averages are lower. Other quarters, averages are higher. So, I adjust for that at the end. Decimal grades are then computed using the equivalencies in the *Time Schedule*, linearly interpolating between letter-grade boundaries.

I am well aware of the significance of assigning a grade below 2.0, in terms of impact on your career here at UWB. I can assure you that I examine *in detail* the performance in this course of each student before assigning a grade below 2.0.

What is the difference between this and grading on a curve? With the latter, the goal is to have $X$% ‘A’s, $Y$% ‘B’s, etc. My way, I would be happy to give out all ‘A’s (if they were earned). FYI, in a “typical” quarter, below 60% might be a ‘D’, 60%–75% a ‘C’, 75%–85% a ‘B’, and above 85% an ‘A’. You may use this as a rough guide; however, if you *really* want to know how you’re doing, please see me. *I reserve the right to adjust these scores to reflect the specifics of assignments, test questions, etc. for each quarter.*

**Class Presentations and Discussions** For this quarter, CSS 343 will not involve formal lectures. Instead, class meetings will include student-led discussions. I will divide the course material up among you, and each of you will be responsible for making a presentation and leading class discussion about a particular topic on particular dates. Please note that *everyone* is responsible for the readings for *every* date. "Class contribution" is evaluated for everyone, not just whoever is making the presentation.

This doesn't mean that if you are presenting you must understand everything before the class — it will be perfectly legitimate to have some open questions coming in. However, as presenter you *are* responsible for very clearly presenting what you do understand and for structuring what you don't understand so that we can discuss it together in class. In other words, "I don't understand *X*," is not a good way to organize a discussion. In particular, I ask that each presenter deliver a set of open-ended questions for everyone else by email to me at least a day before class.

As presenter, you are responsible for going beyond the textbook to gather additional information about the topic at hand. This could be information about major applications of these ideas, products that use the algorithms or data structures, on-line animations of algorithm operation, alternative explanations of concepts, etc. You may use a computer projector, overhead projector, or whiteboard for your presentation (or any combination of these); whichever you think is most appropriate. Please check in with me by email to let me know what you will use so I can make sure that we have everything ready for you. Requests for equipment not already in the classroom may take a day or two to arrange. My evaluation will, in general, be based on the following criteria:

- Are you actively involved in your presentation? Do you show energy and interest, or are you merely "phoning it in"? Does your presentation engender interest and active participation from everyone else, or is it a monologue?
- Is it clear that you've spent time thinking about this subject, or are you merely paraphrasing the textbook? Is everyone else confident in your understanding of the material? For items you don't understand, is it evident that you've looked beyond the textbook and that you've narrowed down the unclear issue to something we can have a focused discussion about?
- Are you organized, or do you jump around from topic to topic without evident purpose? Are your materials easy to understand?
- If you use any graphics, do they contribute to your presentation or are they superfluous or distracting? Is your choice of presentation technology appropriate and to do you make good use of it?
- Are there any spelling or grammatical errors?
- Do you talk to the class, making eye contact with individuals, or do you talk to the screen or board?
- Do you speak clearly and loudly?

**Projects** We will work together as a single group on a large, interesting project or projects. Projects will be selected either by the class, in response to a proposal from students in CSS 490/BBUS 479, or if all else fails by me. If you look over the course topics, you'll see that many of the data structures and algorithms are oriented towards efficient storage and retrieval of indexed data. So, applications that can make use of simple databases are fair game. So are applications that involve things like XML parsing, as both languages and trees are course topics. Path planning and the like, using graphs, would be another possibility. We will have enough people on the project that we will be able to tackle coming up to speed on items beyond the strict scope of the course; this would include such things as learning about XML, developing simple network code using sockets, developing parsers using yacc and lex (or C++ alternatives), etc. So, a simple server program that communicates via sockets with a web server would be a possibility, as would database back ends, business rules in middleware, etc.

As implied above, we will be cooperating with Alan Leong's entrepreneurs workshop, CSS 490/BBUS 479, http://courses.washington.edu/uwben. His class is TTh 5:45–7:50 — right after ours. You are all welcome to visit his class. At the beginning of the quarter, we will get a list of projects proposed by students in his class that are open to us for implementation. If we like one or two of them, then our class could serve as a "prototyping lab" for them. Students from Prof. Leong's class may visit ours from time to time.

I will be an *ex officio* group member, to help out where necessary and to ensure that the project scope stays reasonable for the class. I will attend all group meetings scheduled during class times. You are of course free to schedule group meetings outside of class, too, and by default I will assume that you prefer that these meetings occur away from my "watchful eyes". Since I will be well aware of each group member's responsibilities and performance, I will assign individual grades for project contributions.

**Course "Rules"** The following is a brief summary of the most important things you can do to succeed in this class:

- Read the entire syllabus. Every word.
- You are responsible for making back-up copies of your work. Disk crashes, etc. are *not* acceptable reasons for extensions of assignment due dates.
- Assignments are due when specified. Barring illness or similar extenuating circumstances, please do not attempt to submit amendments, bug fixes, or forgotten material after the fact.
- I may use email to communicate with you. It is your responsibility to assure that email to your UW account reaches and is read by you. Note that the UW course listserve will only accept messages from addresses that are on the official list. So, if you forward your UW email, you will still need to send email messages to the listserve from your UW account. According to my experience in previous quarters, your UW email is almost certainly more reliable than any free account (e.g., Hotmail).

**Special needs** To request academic accommodations due to a disability, please contact Disabled Student Services (DSS) in the Counseling Center, UW1-145, (425) 352-5000, TDD: (425) 352-5303. If you have a documented disability on file with the DSS office, please have your DSS counselor contact me and we can discuss accommodations you might need in class.

**Class attendance** I strongly encourage you to come to all classes, not just those during which you are scheduled to present. Your class contribution grade depends on this. You will be held responsible for all material covered in class, regardless of its presence (or lack thereof) in the textbooks or web site.

**Problems** If you have problems with anything in the course, please come and see me during office hours, or send email to set up an appointment. I want to make you a success in this course. If you have trouble with the assignments, see me before they are due. If you fall behind, it will be difficult to catch up.

## Example Course Schedule

I call the following an "example" course schedule because we may need to modify the order of topics to accommodate whatever project(s) we do. Regardless, this summarizes the topics that will be covered this quarter. The assignment list is largely fictional.

| **Date** | **Topics** | **Reading** | **Assignment** |
|---|---|---|---|
| 1/3 | Course introduction; Inheritance | Weiss, Ch. 4 | |
| 1/5 | Inheritance, cont’d | | Program 1 assigned |
| 1/10 | Trees | Weiss, Ch. 18; Rosen, § 9.1–9.3 | Written HW 1 assigned |
| 1/12 | Trees, cont’d | Weiss, Ch. 18, § 13.1 | Program 1 prelim. spec. & design due |
| 1/17 | Trees, cont’d | | Written HW 1 due |
| 1/19 | Binary Search Trees | Weiss, Ch. 19 | Program 1 due |
| 1/24 | BST, cont’d | | Program 2 assigned |
| 1/26 | BST, cont’d | | |
| 1/31 | Hash tables | Weiss, Ch. 20 | Program 2 prelim. spec. & design due |
| 2/2 | Hash tables, cont’d | | |
| 2/7 | Midterm review | | Program 2 due |
| 2/9 | **Midterm exam** | | |
| 2/14 | Priority queues | Weiss, Ch. 21 | Program 3 assigned |
| 2/16 | Priority queues, cont’d | Weiss, Ch. 14.2 | |
| 2/21 | Graphs | Weiss, Ch. 15; Rosen, Ch. § 8.1, 8.2, 8.3 (pp. 557–9), 8.4 (pp. 567–72), 8.5, 8.6, 9.4, 9.5 | Written HW 2 assigned; Program 3 prelim. spec. & design due |
| 2/23 | Graphs, cont’d | | |
| 2/28 | Boolean Algebra | Rosen, § 10.1–10.3 | Program 3 due; Written HW 2 due; Written HW 3 assigned |
| 3/2 | Boolean Algebra, cont’d | | |
| 3/7 | Modeling Computation | Rosen, § 11.1–11.3 | Written HW 3 due |
| 3/9 | Modeling Computation, cont’d | | |
| 3/14 | **Final exam** | | |