University at Buffalo
Department of Computer Science & Engineering
201 Bell Hall – (716) 645-3180

Syllabus

Please read this sheet carefully, and save it for future reference.

Instructor

Name	Office	Phone 645-3180	Email	Web
Adrienne Decker	130 Bell	Ext. 161	adrienne@cse.buffalo.edu	http://www.cse.buffalo.edu/faculty/adrienne

Course Information

Credit hours: 4
Course Website: http://www.cse.buffalo.edu/faculty/adrienne/FA2008/cse250
Lecture Times:

Monday, Wednesday, Friday		11:00 – 11:50	322 Clemen

Recitation Times:

R1	Tuesday	8:00 – 8:50	21 Baldy
R2	Thursday	11:00 – 11:50	21 Baldy
R3	Thursday	6:00 – 6:50	21 Baldy

Course Description

This course provides a rigorous analysis of the design, implementation and properties of advanced data structures. Topics include order notation and time-space analysis and tradeoffs in list, tree and graph algorithms, and hashing. The course will survey library implementations of basic data structures in a high-level language. Advanced data structure implementations will be studied in detail. The importance of choosing appropriate data structures when solving a problem will be illustrated by programming projects in C++, a high-level object-oriented language different from the language of CSE115-CSE116. There is no expectation that you have C++ programming background prior to this course, though I will assume that you are familiar with basic object-oriented concepts.

This course is a prerequisite for CSE 305 Introduction to Programming Languages, CSE 331 Introduction to Algorithm Analysis and Design, CSE 396 Introduction to the Theory of Computation, and CSE 435 Information Retrieval.

This course adheres to recommendations made in the ACM's CC2001 Computer Science Volume curriculum document for a third semester data structures course. It covers topics from the following knowledge units: DS5 Graphs and Trees, PF3 Fundamental data structures, AL3 Fundamental computing algorithms. It reviews and reiterates concepts from the following knowledge units (due to the change of languages) PF1 Fundamental programming constructs, AL1 Basic algorithm analysis, PL4 Declarations and types, PL5 Abstraction mechanisms, PL6 Object-oriented programming.

Schedule of Topics

The following is a tentative schedule of topics. A more detailed schedule is maintained on the course website and should be checked often for updates.

3 weeks - C++ (syntax review, OO review, templates, namespaces, pointers, make files, debuggers)
2 weeks - Asymptotic notations, properties

1 week - Lists, stacks, queues, deques (STL and analysis)
3 weeks - Trees (e.g. AVL, Red-black, Splay, 2-3, trie)
2 weeks - Priority queues (e.g. binomial, skew, leftist)
2 weeks - Hash tables/hashing
1 week - Graphs (representations, traversals)

Course Objectives

At the end of this course you should be able to perform basic analysis of algorithms, understand how various data structures and algorithms function, be able to implement them in a high-level language, and be able to pick an appropriate data structure or algorithm for a given task.

ABET Program Objectives

Our computer engineering program is accredited by ABET. This course is required of all computer engineering students and has a significant relationship with the following program objectives for computer engineering:

(a) An ability to apply knowledge of mathematics, probability and statistics, computer science and electrical engineering as it applies to the fields of computer software and hardware.
(b) An ability to conduct experiments, as well as to organize, analyze, and interpret data.
(f) An understanding of professional, legal, and ethical issues and responsibilities as it pertains to computer engineering.

This course has a strong relationship with the following program objectives for computer engineering:

(e) An ability to identify, formulate, and solve hardware and software computer engineering problems using sound computer engineering principles.
(k) An ability to use the techniques, skills, and modern hardware and software engineering tools necessary for computer engineering practice.

Prerequisites

You must have passed both CSE 116 and CSE 191 with a grade of C- or better in order to take CSE 250. Your prerequisites will be checked. If you do not have the required prerequisites, you **will be removed from the course**.

Textbooks and Materials

The required textbooks for this course are:

- Elliot B. Koffman & Paul A.T. Wolfgang. 2006. *Data Structures and Design Using C++*, Wiley. (ISBN: 0471467553)
- Stefan Brandle et. al. 2009. *C++ Data Structures*, Jones and Bartlett. (ISBN: 0763755645)

Though you may find the following books useful, they are not required and have not been ordered for the bookstore:

- Nicolai M. Josuttis. 1999. *The C++ Standard Library*, Addison Wesley. (ISBN: 0201379260)
- Stanley B. Lippman and Joseé Lajoie. 2005. *C++ Primer (Fourth edition)*, Addison-Wesley.
- Bjarne Stroustrup. 2000. *The C++ Programming Language: Special Edition (Third edition)*, Addison-Wesley.
- Mark Allen Weiss. 2004. *C++ for Java Programmers*, Prentice Hall. (ISBN: 013919424X)

Additional reading material may be assigned during the course, and will be announced in lecture.

Computing Resources

You will be provided with a CSE undergraduate computing account. You may use the undergraduate lab facilities in Bell. These facilities are available for use as listed on the course website. They are on card-access -

use your UB card to open the door. For your own safety, and to protect the equipment in the lab, do not open or hold the door open in order to allow other people to gain entry to the lab. All students who are authorized to use the lab will be granted card access.

The name of the server that you will be connecting to in the lab will be nickelback.cse.buffalo.edu. You have the ability to connect to timberlake.cse.buffalo.edu remotely from other sites, on or off campus. Both of these machines are file-served from the same machine.

You are expected to become proficient at using the machines in the lab, the Unix system, the C++ compiler, and whatever other software development tools the course requires you to use. It is your responsibility to ensure that any programs you write for this course compile using the C++ compilers installed on the department's machines.

You are also required to read mail sent to your CSE e-mail account. Any e-mail communication that you send regarding this course must be sent from your CSE e-mail account or your UB e-mail account. Under no circumstances will e-mail from non-UB accounts be acknowledged or answered. You must include an informative subject line in all e-mail, and include your full name in any e-mail correspondence.

All e-mail that we send in reply to your e-mail will be sent to the address from which you sent your e-mail. Our feedback on materials you hand in electronically will be sent to your CSE e-mail account only. Since you may request re-grades of work only within a set period from the time that the feedback was provided to you, it is in your best interest to read your CSE e-mail account on a daily basis.

Course Organization

The course has both a lecture component and a recitation component. Each component plays a role in helping you achieve the objectives of the course. If you do not participate fully in both you should not expect to do well in the course.

Lectures

The conceptual and theoretical course content will be delivered primarily in the lectures, complemented by readings from the text books. You must review readings prior to attending a lecture, and you are expected to review the readings again, along with any notes you took, after the lecture.

Some of the topics will be difficult. It is therefore absolutely essential that you ask questions whenever something is said which you do not understand.

You are expected to attend all lectures. If you are unable to attend a lecture because of sickness or similar reasons, make sure you get the notes from a classmate. If you are out of class for an extended period of time because of sickness, notify your instructor as soon as possible, and see your instructor immediately upon your return in order to determine how to catch up. If you have missed a significant portion of the semester due to illness, it is recommended that you resign from the course.

Recitations

The recitations are an integral part of the course. They will cover C++ programming in detail. Attendance in recitation will therefore be critical for your ability to complete the programming projects.

The recitations may review and extend lecture material and are also an excellent forum for asking more individual questions about the course material than can typically be addressed in lecture. Some material needed to do the programming projects will be covered only in recitation. Any homework will be collected and returned in recitation. Quizzes are returned in recitation. Attendance in recitation is expected.

Recitations do not meet in the first week of classes.

Time outside of class

Office hours

Office hours offer you the opportunity to ask more individual questions about the course material than can typically be addressed in lecture. Both the instructor and the teaching assistants have scheduled office hours. Office hours are held on a first-come first-served drop-in basis. No appointment is necessary to attend office hours. Be aware that office hours become increasingly busy the closer it is to a project deadline. Plan your use of office hours accordingly. Individual appointments may be arranged, if needed, as schedules allow.

Study time

In this course, as in any course, you are expected to put in additional time beyond the scheduled class times. Professors generally expect that for each credit hour a class carries a typical student will put in 2 - 3 hours of time each week outside of class. Since this is a 4 credit course that translates into 8 - 12 hours of time outside of lecture and recitation times, each week. During this time you should review your lecture notes, attend office hours as needed, get hands-on practice applying the concepts and theoretical constructs discussed in class, and possibly arrange to meet in small groups to study or review the concepts from class. As a rough guide, you should expect to spend at least the following time working on this course, each week:

- Lectures: 3 hours
- Recitation time: 1 hour
- Programming assignments: 5 hours
- Individual study: 4 hours

Course evaluation

The following indicates the grade breakdown which will be used in assigning grades in the course. The right is reserved to make small adjustments to the breakdown if it is necessary.

Exam component (50% of final course grade)

There will be four in-class examinations and one final examination at the end of the term. The dates for the midterm exams will be posted to the course website. The final examination will be given on a date to be specified by the University. **Do not** make travel plans for times during the examination period until the final examination schedule has been posted.

If you miss an examination because of sickness or similar reasons, visit a physician and obtain a note detailing the period during which you were medically incapable of taking the exam. Notify your instructor immediately via e-mail or telephone (voice mail) if you are going to miss an exam, before the exam takes place unless medically impossible. See your instructor as soon as you return to class.

If you miss an examination without a valid excuse, you will receive a zero grade for that examination.

No make-up examinations will be available without a valid excuse. You must bring a valid form of picture ID with you to each examination (a UB Card will suffice).

The lowest of your four in-class exam grades will be dropped if you take each of the four in-class exams.

There are two options for calculating your score for the exam component of the course. Under the first option the in-class exams count for 25% of your grade, while the final exam counts for 25%. Under the second option the final exam counts for 50% of your grade. The option which gives you the highest score in the course will be used automatically.

You must attempt all the in-class exams in order for the final-exam only option to be available to you.

The motivation for having two grading options available is to ensure that you are not penalized if you had a rough start in the course, but managed to do really well on the final exam. If you do poorly on one or more of

the in-class exams, you can still do well in the course by demonstrating that you have learned the material on the final exam.

The following table summarizes the grading of the exam component of the course:

	Option #1	Option #2
In-class exams	25%	0%
Final Exam (Cumulative)	25%	50%

A necessary but not sufficient condition for receiving a passing grade in the course is having a passing exam component grade.

Homework and Lab work component (20% of final course grade)
There will be regular homework and "labwork" assignments throughout the semester. Late submissions of these assignments **will not be accepted under any circumstances**. Early submissions will not be awarded any bonus. Your lowest homework/labwork grade will be dropped from final course grade calculations. All homeworks and labworks will count equally.

You must have a passing average on the homework/labwork component of the course in order to pass the class.

Project component (30% of final course grade)
There will be programming projects. The purpose of these is to reinforce and deepen your understanding of the broader concepts discussed in class through application of those concepts to concrete problems. The programming projects are designed to give you hands-on experience analyzing problems, developing solutions to them, and implementing these solutions in C++. The programming projects also serve to give you feedback on your understanding of the material. I expect that we will have two-three programming projects, weighted equally. It is your responsibility to ensure that any programs you write for this course compile using the C++ compilers installed on the department's machines. Submissions which do not compile will not be graded.

You must have a passing average on the project component of the course in order to pass the class.

Early policy for programming project submissions
Any programming project submission which occurs before the due date is considered early, and will have a 2.5% bonus (of the maximum score obtainable) added per full day early (24 hours), up to a maximum of 10%.

Late policy for programming project submissions
Any programming project submission which occurs after the due date is considered late, and will have a 25% penalty (of the maximum score obtainable) imposed per day (24 hours), or portion thereof, late. A submission more than three days late (i.e. four or more days late) will therefore be awarded no points.

When calculating final course grades, I will "forgive" two days of programming project late penalties. I will "forgive" the two late penalties which affect your grade the most. For example, if all of your submissions except one are on time, and the late submissions is two days late, it counts for full credit. As another example, if the late submissions is three days late, but all your other submissions are on time, the late submission counts as one day late. Unused late days do not benefit you.

Re-grading
If you have a question about the grading of any piece of work, first consult with the teaching assistant who graded your work. If you cannot resolve your questions with the teaching assistant, you should consult with the instructor of the course.

Any questions about the grading of a piece of work must be raised within one week of the date that the work was returned by the teaching assistant or the instructor. In other words, if you do not pick up your work in a timely fashion, you may forfeit your right to question the grading of your work.

Incomplete (I) grades

We will follow the UB Undergraduate Catalog Statement on Incomplete Grades, found in the Undergraduate Catalog.

Generally, incomplete ("I") grades are not given. However, very rarely, circumstances truly beyond a student's control prevents him or her from completing work in the course. In such cases the instructor can give a grade of "I". The student will be given instructions and a deadline for completing the work, usually no more than 30 days past the end of the semester. University and department policy dictate that "I" grades can be given only if the following conditions are met:

- An Incomplete will only be given for missing a small part of the course.
- An Incomplete will only be given when the student misses work due to circumstances beyond his/her control.
- An Incomplete will only be given when the student is passing the course except for the missed material.
- An Incomplete is to be made up with the original course instructor within the time specified by the appropriate University regulation (see appropriate document above), and usually within the following semester.
- An Incomplete will not be given to allow the student to informally retake the entire course, and have that grade count as the grade of the original course.

Incompletes can not be given as a shelter from poor grades. It is your responsibility to make a timely resignation from the course if you are doing poorly for any reason. The last day to resign the course is Friday, November 7th.

Letter grades

The following table indicates the number to letter grade mapping I will use to assign final grades at the end of the course. The Grade points column is included for your convenience only, and is not official information. The official mapping can be found in the Undergraduate Catalog.

Percentage score	Letter grade	Grade points
90-100	A	4.0
85-89	A-	3.67
80-84	B+	3.33
75-79	B	3.0
70-74	B-	2.67
65-69	C+	2.33
60-64	C	2.0
55-59	C-	1.67
50-54	D	1.0
0-49	F	0.0

General Notes

If you don't understand something covered in class, ask about it right away. The only silly question is the one which is not asked. If you get a poor mark on an assignment, quiz, or exam, find out why right away. Don't wait a month before asking. The instructor and teaching assistants are available to answer your questions. Don't be afraid to ask questions, or to approach the instructor or T.A. in class, during office hours, or through e-mail.

This course is intended to be hard work, but it is also intended to be fun. Play with the computer, and have fun with the neat and elegant programming ideas covered in this course. We think computer science is interesting and exciting, and we want to convince you of this. Work hard, but have fun!

Disabilities

If you have a diagnosed disability (physical, learning, or psychological) that will make it difficult for you to carry out the course work as outlined, or that requires accommodations such as recruiting note-takers, readers, or extended time on exams or assignments, you must consult with the Office of Disability Services (25 Capen Hall, Tel: 645-2608, TTY: 645-2616, Fax: 645-3116,
http://www.student-affairs.buffalo.edu/ods/).

You must advise your instructor during the first two weeks of the course so that we may review possible arrangements for reasonable accommodations.

Counseling Center

Your attention is called to the Counseling Center (645-2720), 120 Richmond Quad. The Counseling Center staff are trained to help you deal with a wide range of issues, including how to study effectively and how to deal with exam-related stress. Services are free and confidential. Their web site is http://www.student-affairs.buffalo.edu/shs/ccenter/

Distractions in the Classroom - Behavioral Expectations

The following is the text of a policy adopted by the Faculty Senate. You are expected to know and adhere to this policy.

OBSTRUCTION OR DISRUPTION IN THE CLASSROOM – POLICIES UNIVERSITY AT BUFFALO

To prevent and respond to distracting behavior faculty should clarify standards for the conduct of class, either in the syllabus, or by referencing the expectations cited in the Student Conduct Regulations. Classroom "etiquette" expectations should include:

- Attending classes and paying attention. Do not ask an instructor in class to go over material you missed by skipping a class or not concentrating.
- Not coming to class late or leaving early. If you must enter a class late, do so quietly and do not disrupt the class by walking between the class and the instructor. Do not leave class unless it is an absolute necessity.
- Not talking with other classmates while the instructor or another student is speaking.
- If you have a question or a comment, please raise your hand, rather than starting a conversation about it with your neighbor.
- Showing respect and concern for others by not monopolizing class discussion. Allow others time to give their input and ask questions. Do not stray from the topic of class discussion.
- Not eating and drinking during class time.
- Turning off the electronics: cell phones, pagers, and beeper watches.
- Avoiding audible and visible signs of restlessness. These are both rude and disruptive to the rest of the class.
- Focusing on class material during class time. Sleeping, talking to others, doing work for another class, reading the newspaper, checking email, and exploring the internet are unacceptable and can be disruptive.
- Not packing bookbags or backpacks to leave until the instructor has dismissed class.

Academic Integrity

Source: http://www.cse.buffalo.edu/academics-academic integrity.shtml

The academic degrees and the research findings produced by our Department are worth no more than the integrity of the process by which they are gained. If we do not maintain reliably high standards of ethics and integrity in our work and our relationships, we have nothing of value to offer one another or to offer the larger community outside this Department, whether potential employers or fellow scholars.

For this reason, the principles of Academic Integrity have priority over every other consideration in every aspect of our departmental life, and we will defend these principles vigorously. It is essential that every student be fully aware of these principles, what the procedures are by which possible violations are investigated and adjudicated, and what the punishments for these violations are. Wherever they are suspected, potential violations will be investigated and determinations of fact sought. In short, breaches of Academic Integrity will not be tolerated.

Departmental Statement on Academic Integrity in Coding Assignments and Projects

The following statement further describes the specific application of these general principles to a common context in the CSE Department environment, the production of source code for project and homework assignments. It should be thoroughly understood before undertaking any cooperative activities or using any other sources in such contexts.

All academic work must be your own. Plagiarism, defined as copying or receiving materials from a source or sources and submitting this material as one's own without acknowledging the particular debts to the source (quotations, paraphrases, basic ideas), or otherwise representing the work of another as one's own, is never allowed. Collaboration, usually evidenced by unjustifiable similarity, is never permitted in individual assignments. Any submitted academic work may be subject to screening by software programs designed to detect evidence of plagiarism or collaboration.

It is your responsibility to maintain the security of your computer accounts and your written work. Do not share passwords with anyone, nor write your password down where it may be seen by others. Do not change permissions to allow others to read your course directories and _les. Do not walk away from a workstation without logging out. These are your responsibilities. In groups that collaborate inappropriately, it may be impossible to determine who has offered work to others in the group, who has received work, and who may have inadvertently made their work available to the others by failure to maintain adequate personal security In such cases, all will be held equally liable.

These policies and interpretations may be augmented by individual instructors for their courses. Always check the handouts and web pages of your course and section for additional guidelines.

Departmental and Course Policy on Violations of Academic Integrity

If, after following the procedures required by the University for investigation of suspected breaches of academic integrity, a student is found guilty, the policy of the department of Computer Science & Engineering is that the student minimally receive a grade of F in the course.

University at Buffalo
Department of Computer Science & Engineering

I, _____(PRINT name), acknowledge that I have read and understood the syllabus for this course, CSE 250 Data Structures.

I also acknowledge that I understand the definition of academic integrity as outlined in the syllabus, and that I will minimally receive a grade of F in the course if I am found to have breached academic integrity.

I also understand that I am required to have successfully completed all of the listed prerequisites for this course with a minimum grade of C-. I understand that if I do not meet the prerequisites that I may be dropped from the course by the department.

Signature: _____ Date: _____