

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	Email	Web
Adrienne Decker	130 Bell	645-3180 x 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/FA2006/cse116>

Section A - Monday, Wednesday, Friday 11:00 – 11:50 107 Talbert

Recitation Times:

A1	Tuesday	2:00 – 3:50	21 Baldy
A2	Wednesday	12:00 – 5:50	21 Baldy
A3	Thursday	2:00 – 9:50	21 Baldy

Course Description

This course is a continuation of CSE 115/503, in which heavy emphasis is placed on abstract datatypes (ADT's) and object-oriented methodology. You will be expected not only to understand abstract data types, but also how to design and implement robust ADT's using a modern object-oriented programming language (Java). Topics such as encapsulation, polymorphism, interfaces, inheritance and composition will be emphasized. Essential topics to be integrated in this framework include the use of recursion, references, linked structures including linked lists, binary trees, stacks, queues, and other advanced data structures and algorithms, including advanced searching and sorting algorithms.

The analysis of algorithm complexity (O-notation) will be introduced. In the course of working on a semester-long software project you will be introduced to a software development methodology, select software development tools, software documentation and teamwork. You will work as part of a software development team; your team will be required to give a brief oral presentation of and demo your course project.

More detailed information on lecture topics and required readings is available on the course web site.

This course adheres closely to the recommendations of ACM's CC2001 curriculum document for a second semester introductory course. It covers topics from the following knowledge units: PF3 Fundamental data structures, PF4 Recursion, PF5 Event-driven programming, AL1 Basic algorithmic analysis, AL2 Algorithmic Strategies, AL3 Fundamental computing algorithms, AR2 Machine level representation of data, PL2 Virtual machines, PL4 Declarations and types, PL5 Abstraction mechanisms, PL6 Object-oriented programming, HC2 Building a simple graphical user interface, SE1 Software design, SE2 Using APIs, SE3 Software tools and environments, SE4 Software processes, SE6 Software validation and SE7 Software evolution.

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.

1 week – Arrays and Iterators

1 week – Recursion

2 weeks – Lists

1 week – Stacks & Queues

2 weeks – Trees (Binary Search Trees)

2 weeks – Big-oh notation, asymptotic analysis, searching, sorting

2 weeks – Graphics, Exceptions, Handling Files

Course Objectives

The main objectives of the this course, together with the primary mechanisms by which these objectives will be met, are described below.

- One objective is to introduce fundamental data structures, such as stacks, queues and trees, as well as their implementation. The lectures will present and discuss these data structures, their uses and their implementation. You will implement and use these data structures as part of your project.
- Another objective is to introduce algorithm analysis using O-notation. The lectures will present and discuss the O (big-Oh) notation for describing an algorithm's resource (space or time) usage. You will analyze the performance of data structures used as part of your project theoretically and empirically.
- A third objective is to provide you with skills and tools needed to design and implement a software application of moderate size, using object-oriented design techniques in a team environment. You will gain experience by working on a large team-based project throughout the semester. The project will be designed and built incrementally as the semester proceeds.

Prerequisites

You must have passed CSE 115 with a minimum grade of C-, or have an equivalent background and receive permission from the instructor.

Textbooks and Materials

The required textbooks for this course are:

- Michael Main. 2006. Data Structures & Other Objects Using Java (3rd ed), Addison Wesley. (ISBN: 0-321-37525-4)
- Cay Horstmann. 2006. Object-Oriented Design and Patterns (2nd ed), John Wiley & Sons, Inc. (ISBN: 0-471-74487-5)

Though you may find the following book useful, it is not required and has not been ordered for the bookstore:

- Cay S. Horstmann & Gary Cornell. 2005. Core Java 2 Volume I – Fundamentals, seventh edition, Prentice Hall (ISBN: 0-13-148202-5)

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 Baldy. 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 yeager.cse.buffalo.edu. You have the ability to connect to yeager remotely from other sites, on or off campus.

You are expected to become proficient at using the machines in the lab, the Unix system, the Java compiler as integrated with Eclipse IDE, 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 Java 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 lab (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.

Labs

Labs provide a guaranteed weekly meeting time for project teams. This will be an essential time for your and your project team to discuss and plan your project work each week.

The labs may also review and extend lecture material and are also an excellent forum for asking more individual questions or questions your team may have about the course material than can typically be addressed in lecture. Some material needed to do the programming projects will be covered only in lab. Attendance in lab is expected and will be taken every week.

Labs do not meet in the first week of classes.

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 (48% of final course grade)

There will be ~~one~~ **two** midterm examinations and one final examination at the end of the term. The midterm exams will be held on ~~Monday, October 23~~, **Wednesday, October 4 and Wednesday, November 8**. **The two midterms exams will be weighted 14% each and the final exam will be weighted 20% of the overall course grade.** 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).

In order for you to receive a passing grade in this course, you must have a passing exam average.

Attendance component (10% of final course grade)

Attendance is required and will be taken at all class meetings (lectures and labs). Failure to attend class meetings could have a negative impact on your grade. It is your responsibility to ensure that you sign the attendance sheet at every class meeting. You will be allowed 3 missed lectures and 1 missed recitation “free” during the semester (ie – without penalty to your grade). After that, each lecture or lab you miss will incur a 2% reduction on your attendance grade. After you have missed an additional five lectures or labs, you will have earned a 0% for your attendance grade.

Quiz component (4% of final course grade)

During the semester, there will be in-class quizzes of the announced or unannounced variety. There will be no make-ups for missed in-class quizzes given. If you miss an in-class quiz, you will earn a 0 for that quiz.

Project component (38% of final course grade)

This course has a semester-long team programming project. The project is developed incrementally in three stages throughout the term. Designing and implementing this project will give you experience working with a moderately large piece of software, which is very important in order to develop sound design and programming skills and habits. You will work as a member of a small team to complete the course project. Your scheduled recitation time is an essential meeting time for you and your team members.

You will make three submissions for the long project, one at the end of each stage. Stage 1 is worth 8%, stage 2 is worth 11% and stage 3 is worth 14%. Each stage will require submission of both software and written documentation.

You and your teammates must also give an oral presentation/demonstration of your project, worth 5%. All team members must participate in the presentation.

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 33.33% penalty (of the maximum score obtainable) imposed per day (24 hours), or portion thereof, late. A submission more than two days late (i.e. three or more days late) will therefore be awarded no points. However, when calculating overall course grades I will “forgive” three days of programming project late penalties. I will “forgive” the three late penalties which affect your grade the most. For example, since stage 3 is worth 15% of your grade whereas stage 1 is worth 8%, a late day on stage 3 affects your grade more than one on stage 1. Unused late days do not benefit you.

Peer evaluation

The programming project is a team-based project. For each stage submission each team member will evaluate the contribution of each member of their team, including themselves. The peer evaluation produces a multiplier for each team member. The team member's score is the submission's overall score multiplied by the individual multiplier. More details on this peer evaluation procedure will be handed out during the semester.

I reserve the right to ignore the peer evaluations in cases where I feel they are not an accurate reflection of a student's contribution to the project

Regrading

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.

Newsgroup

There is a newsgroup, sunyab.cse.116, for this course. You must learn how to read news and subscribe to this newsgroup. You are expected to read the newsgroup on a daily basis. There will often be important material posted there, such as supplementary course notes, homework and sample exam questions, and occasionally late breaking news. You may post general course related articles to the newsgroup. Use discretion in posting articles related to homework assignments: when in doubt, e-mail the T.A. or instructor first.

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 for all students is Friday, November 10th.

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 on 5/2/2000. 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](http://www.cse.buffalo.edu/academics-academic%20integrity.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 116 Introduction to Computer Science for Majors II.

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: _____