

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.

Instructors

Name	Office	Phone	Email	Web
Adrienne Decker	130 Bell	645-3184	adrienne@buffalo.edu	http://www.cse.buffalo.edu/faculty/adrienne

Course Information

Credit hours: 4

Course Website: <http://www.cse.buffalo.edu/faculty/adrienne/SP2011/cse115>

NOTE: UBLearns will only be used for the posting of grades. All other course materials will only be available from the website given above.

Lecture Times:

Monday, Wednesday, Friday 3:00 - 3:50 112 Norton

Recitation Times (all recitations meet in 21 Baldy):

A1	T	3:00 - 4:50	A4	F	12:00 - 1:50
A2	W	12:00 - 1:50	A5	W	5:00 - 6:50
A3	R	9:00 - 10:50			

Course Description

Provides the fundamentals of the field to computer science and computer engineering majors, introducing students to algorithm design and implementation in a modern, high-level programming language. Emphasizes problem solving by abstraction. Topics include object-oriented design using a formal modeling language; fundamental object-oriented principles such as classes, objects, interfaces, inheritance and polymorphism; simple event-driven programming; data types; variables; expressions; basic imperative programming techniques, including assignment, input/output, subprograms, parameters, sequencing, selection and iteration; the use of aggregate data structures, such as arrays or more general collections; simple design patterns.

This course adheres closely to the recommendations of ACM's CC2001 curriculum document for a first semester introductory course. It covers topics from the following knowledge units: PF1 Fundamental programming constructs, PF2 Algorithms and problem-solving, PF5 Event-driven programming, PL1 Overview of programming languages, PL4 Declarations and types, PL5 Abstraction mechanisms, PL6 Object-oriented programming, HC2 Building a simple graphical user interface, SP1 History of Computing, SE1 Software design, SE2 Using APIs, SE3 Software tools and environments.

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. **(g)** An ability to effectively communicate technical information in speech, presentation, and writing. **(j)** A knowledge of contemporary issues.

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.

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.

2 weeks - Objects and relationships	1 week - Programming with graphics
2 weeks - Methods, parameters, and encapsulation	2 weeks - Arithmetic, selection, iteration
3 weeks - Inheritance and polymorphism	2 weeks - Collections and iterators

Learning Objectives

By the end of this course, students will be able to:

- Create a program that uses object-oriented design and its key concepts of encapsulation, inheritance, and polymorphism
- Create a program that has a graphical user interface (GUI) and components that the user of the program can interact with (menus, buttons, etc)
- Use an integrated development environment (Eclipse's JDT) to perform typical program editing and navigation tasks
- Identify where and when to use constructs such as selection and iteration in their programs
- Create and use various types of collections where appropriate in programs
- Create and use iterators where appropriate in programs

Prerequisites

There are no prerequisites for this course. You do not need previous programming experience to take this course. However, it is expected that you have *used* a computer before.

Textbooks and Materials

The required textbooks for this course are:

- Carl Alphonse & Adrienne Decker. *CSE 115 Course Notes* [Available on a chapter-by-chapter basis on course website]

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 21 when accessible and when not in use for recitations for any other course. A schedule of the times Baldy 21 is not in use is available on the course website. The lab is 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 `styx.cse.buffalo.edu`. You have the ability to connect to `timberlake.cse.buffalo.edu` remotely from other sites, on or off campus.

You are expected to become proficient at using the machines in the lab, the Linux operating system, the Java compiler, Eclipse's Java Development Tools (JDT), the CVS versioning system, 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 compiler installed on the department's machines. Information about the CSE computing environment can be found at wiki.cse.buffalo.edu/services/

You are also required to read mail sent to your UB e-mail account. Any e-mail communication that you send regarding this course must be sent from a 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. It is also most helpful if you include in the subject or the message body, the number of the course you are taking (CSE 115 or CSE 503) because in a typical semester I teach more than one course.

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 available either through Web-CAT,

web-cat.cse.buffalo.edu:8443/Web-CAT

or by e-mail sent to your UB e-mail account. Announcements and other messages may also be sent via email using the facilities for sending mass email available in UBLearns. Since you may request re-grades of work only within a set period from the time that the feedback was provided to you, and course announcements could provide you with important information, it is in your best interest to read your UB e-mail account on a daily basis.

Course Organization

The course has both a lecture component and a recitation (lab) 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.** Outside of the scheduled course times, both office hours and your own study times are critical components of 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. In each recitation section, the TAs will cover material pertinent to the current assignment. The recitation sections are held in the computer lab in Baldy 21 and you will have the opportunity to work on your assignments during recitation and ask questions of the TA about your work. Attendance in recitation will therefore be critical for your ability to complete the programming projects.

The recitations may also 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. Attendance in lab is expected and will be taken every week. Attendance in recitation will count towards your programming project grades.

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.

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 (56% of final course grade)

There will be five examinations given during the semester and one final examination at the end of the term. The in-class exam schedule will be posted on 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 five in-class exam grades will be dropped if you take each of the five 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 28% of your grade, while the final exam counts for 28%.

Under the second option the final exam counts for 56% of your grade. The option which gives you the highest score in the course will be used automatically.

You must attempt all in-class exams in order for the final-exam only option to be available to you. If you do not write all in-class exams, you cannot make use of the final-exam only option.

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 the in-class exams, you can still do well in the course by demonstrating that you have learned the material on the final exam. Of course, if you do poorly on the in-class exams, this means you are playing without a safety net.

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

	Option #1	Option #2
In-class exams	28%	0%
Final Exam (Cumulative)	28%	56%

A necessary but not sufficient condition for receiving a passing grade in the course is having a passing exam average. This means, in order to pass the course you must not only have an overall course grade that is passing, you must also have a passing average on the exams. Therefore, if your exam average is failing, you will fail the course as a whole, regardless of how well you have done in the rest of the course.

Project component (44% of final course grade)

There will be regular 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 Java. The programming projects also serve to give you feedback on your understanding of the material.

We expect that we will have eight programming projects, weighted as described in the following table.

Lab	1	2	3	4	5	6	7	8
Weight	4%	5%	5%	5%	6%	6%	6%	7%

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. **Submissions which do not compile will not be graded, and will receive an automatic grade of 1 point.**

A necessary but not sufficient condition for receiving a passing grade in the course is having a passing lab average. This means, in order to pass the course you must not only have an overall course grade that is passing, you must also have a passing average on the labs. Therefore, if your lab average is failing, you will fail the course as a whole, regardless of how well you have done in the rest of the course.

Late policy for programming project submissions

Any programming project submission which occurs after the due date is considered late, and will have a 50% penalty (of the maximum score obtainable) imposed per day (24 hours), or portion thereof, late. In no case shall a lab grade be less than zero. A submission more than one day late (i.e. two or more days late) will therefore be awarded no points. No exceptions will be made to this policy unless written notice of a documented medical issue, family emergency, or sanctioned university event can be provided. It is up to the discretion of the instructor how to deal with documented events such as those mentioned previously.

Re-grading

If you have a question about the grading of any piece of work, first consult with the staff member who graded your work. If it was a teaching assistant who graded your work and you are unable to resolve your questions with that person, 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, March 26th**.

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.

Recall, in order to pass the course you must not only have an overall course grade that is 55% or higher, you must also receive an exam average that is above 55% and a lab average that is above 55% or you will receive an automatic grade of F.

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
0-54	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](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.