Course Policies

Instructor: S. Chaiken
Office Hours: T, Th: 2:40PM-4:15
LI 96H, 442-4282 Wed: 9:30-11:00AM
sdc@cs.albany.edu
other times drop-in if I'm not busy

Prerequisites: CSI201 or equivalent CS1 course in fundamental programming (variables, arrays, control statements, methods, classes, basic problem solving) with Java, including writing, testing and debugging programs. A prerequisite examination will be administered at the beginning of the course and students who do not have the prerequisites may be deregistered. Various “Book Classes” from Guzdial and Ericson, and applications, covered in Albany’s CSI201 of will be used, so students whose CS1 course did not cover these are expected to take special effort to catch up.

Many find this course fast paced and challenging. If you feel at all unprepared for this course, especially if you did not take CSI201 recently at Albany, you should participate in CSI201 (not necessary to register) and decide during the next two weeks between taking this course and CSI201.

Required Texts:

1. Mark Guzdial and Barbara Ericson, “Problem Solving with Data Structure Using Java, a Multimedia Approach, 1st, Prentice Hall 2011. Java libraries (“Book Classes”) and example programs supporting this text are explained in the text. We will continue from those authors’ “Introduction to Computing and Programming in Java: A Multimedia Approach” textbook used at Albany for CSI201.

   Reading assignments will be assigned for which in-lecture reading and programming QUIZZES will be given. I expect every student will at least have access to the textbook.

   Homework Programming Assignment One, DUE Monday, 12:00 Noon. Make a complete Java application program that prints something when it is run, make sure it compiles and runs properly, and then submit the source file, a .java file, on Blackboard. If you don’t know how to do this, find out from a classmate who as already done this at UAlbany. If you want to, make the program more interesting but limit it one class and to 20 lines of code.

2. Supplemental readings posted on the Web will be assigned.

3. Your CS1 Java textbook and/or other brief references for Java programming fundamentals. Addison-Wesley’s “Java Backpack Reference Guide” by DePasquale is an excellent reference for the important details one must use but should not remember when one is engaged in programming. It is an inexpensive addon in the textbook package I specified to the bookstores.

4. Project assignments are software specifications: Every word on project assignments, and on specified readings they rely on, is required reading. The professor and TAs will ask you about these readings when addressing your difficulties.

5. Course lecture notes and external Web references provided during the course.
1 Learning Objectives

1. To demonstrate fluency and problem solving skills in computer programming at levels expected during the second semester of University Computer Science major study. In other words: Given a sufficiently detailed description of what a computer can do, the student shall write a program that makes the computer do it.

2. Apply to programming and analytical problems the principles, practices and elementary analyses of some specific data structures and program logic control constructions.

The (more or less non-obvious) data structures are well-known and understood by computer scientists, professional programmers and software designers. The power, efficiency and, for some beginners, the learning difficulty of these linked and/or arrayed data structures is a consequence of the fact that they contain data that refers to data.

Within these data structures, there are pointer or reference and/or array index DATA whose purpose in to locate OTHER DATA. (The other data is accessed, referred to, or retrieved in short, constant time; no searching is needed.) The other data is often structured and includes its own pointers or references and/or indexes to function in the same linked or arrayed manner. Additional efficiency results from other contraints on data organization, for example, the arrays, lists or trees being sorted or ordered.

We will also introduce examples of similar data organization from the Web, the Internet and computer operating systems/environments.

Students will analyze and write code whose logic is organized into methods (or functions) calling one another, including recursive situations in which a method calls the same named method directly or indirectly before the original method call returns.

Recursion is a powerful property defining, problem solving, algorithm structuring and computer coding technique. Recursion means to define something in terms of itself in a way that avoids the philosophical problem of circularity. (By the way, abstract mathematical approaches to recursion are taught in Albany’s CSI210 course.)

3. Students will complete programming projects and to solve problems on paper pertaining to trees and their relation to various hierarchically organized data and mathematical models. These include scene graphs, taxonomies, nested expressions, introductory context free grammars, and recursion.

Important but secondary learning objectives include applying principles of object oriented design and programming in projects, as supported by some features of Java. They also include beginning proficiency with computing environments including a command line shell controlled Unix system that expose and enable programmers to control, customize and script the steps relevant to their tasks.

A. Laboratory Exercises and Technology Tutorials: The schedule for lab sessions and staff office hours will be linked from the course web page.

The lab periods will be used for learning and discussion exercises conducted by the assistants. Some labs may feature self-paced tutorials. Accomplishments in every lab will be assessed with the assistant checking your work and/or material submitted on paper or electronically to Blackboard.

You must register for some lab session. If the session is overfull, the professor will accomodate you entering a different session. If you cannot come to your registered lab period in any single
week, or you need more time to complete its requirements, you can go to a different one that week so long as there is physical room (which there is likely to be.)

Quizzes will sometimes be given in lectures to monitor student preparation, progress, learning needs and attendance. Points for quiz answers will be included in the grade.

2 Lab meetings will begin Monday (1/24).

But, we will hold an orientation and catch-up session this Friday 1/21, 11:30PM in HU-25, particularly for transfer students. But, all who want a refresher on writing Java programs in the CS Dept. Lab are welcome.

IMPORTANT: You must have and know your UAlbany NetId and password for lab and homework. If you do not have these, finish your UA registration as soon as possible and get them.

Additional help/office hour resources will be posted on the course web page as soon as they are available.

B. Evaluation:

<table>
<thead>
<tr>
<th>Evaluation Type</th>
<th>Date/Time</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>Mar. 10, 2011 (Thurs)</td>
<td>20%</td>
</tr>
<tr>
<td>Final</td>
<td>May 11, 2011 (Wed)</td>
<td>30%</td>
</tr>
<tr>
<td>Lecture Quizzes</td>
<td></td>
<td>10%</td>
</tr>
<tr>
<td>Lab. Exer.,</td>
<td></td>
<td>10%</td>
</tr>
<tr>
<td>Homework Projects (6 or 7)</td>
<td></td>
<td>30%</td>
</tr>
</tbody>
</table>

Final letter grades will be based on cutoffs applied to your score computed as above, with the professor reserving the right to assign higher grades based on a record of substantial improvement.

The grading scale will be scale so that C (2.0) indicates the minimum necessary preparation for passing CSI333 (Programming at the Hardware/Software Interface or “Assembly Language”, the 3rd Albany CS Major programming sequence course). Similarly for CSI311 (Principles of Programming Languages), if you also get a C or better in CSI210.

Estimated grades based on the midterm exam score will be announced with the midterm results.

The “C” (2.0) or better grade in CSI310 is a prerequisite for the two required CS major courses CSI333 and CSI311. If your are a CS major, CSI310 is your most important course so far and you should aim for an “A”. (Ask any professional programmer.) If you get a B, that is fine. If not, consider switching majors. If you get below C, you are probably not in the right major: Your choices are to switch or try CSI310 again.

CSI310 is a required course for the CS minor, and it is an elective for ISP and a few other majors. For these academic programs, just passing, not a C or better grade, is required.

Details regarding the exams will be announced later. Sample questions will be posted on the course web site.

C. Homework Programming Projects: There will be eight to ten homework programming project assignments. Some will be continuations of previous ones.

Submissions that do not compile get ZERO points, automatically! See the outline for doing incremental software development included in the project 1 assignment. There is
No excuse for turning in program code with syntax errors in a 2nd semester major course! It is expected that you will start the projects early enough so that you can get any help you may need to get the syntax right.

Although the programming work can be done on your own computers, and this is encouraged, problems (hardware, software, network access) with your system will not be accepted as excuses for late or missing programming project or lab completion work.

Programming guidelines and submission information will appear in a separate handout.

D. Policy on Cheating:

1. Cheating in an exam will result in an E grade for the course. Further, the students involved will be referred to the University Judicial System.

2. The code and any written reports or answers for homework programming assignments; and written answers for lab exercises must be written by yourself. You are welcome to discuss the class material, the problems and ideas for solutions; but each person is expected to write the code and answers he or she submits independently, without copying.

Cheating in a programming assignment or lab exercise will result in a ZERO for that requirement for all the students involved. Students who cheat in two or more programming assignments will receive an E grade for the course.

A report of the every cheating incident will also be made to the Office of Undergraduate Studies in accordance with the University regulations concerning “Penalties and Procedures for Violations of Academic Integrity” in the Undergraduate Bulletin.

We may do automated code comparisons between submissions of current students together with submissions from prior course offerings to detect copying.

3. Violations of Computer and Network Usage policy on laboratory systems are regarded as academic integrity violations, like cheating.

E. Policies on Computer and Network Usage:

1. Attempts to use any University computers or networks in violation of University, course and laboratory policies may result in account suspension, course failure or referral to University disciplinary action. The implications of the responsible use requirements of shared computer and network infrastructure is an element of this course’s curriculum. Course failure or grade reduction can be imposed by the instructor as a sanction against the violation of laboratory rules and procedures. Failure to comply with instructions given to you by system administration or course instructional team staff members in regard to Lab usage is disruptive and is grounds for sanction. Willful illegal, malicious or disruptive use, or attempts to disguise one form of computation as another will be taken particularly seriously.

If you want to try some kind of computer stress testing experiments, talk to the professor and you may get permission to do it responsibly on a special, isolated system!

You are also warned against disclosing your password to anybody either willfully or by accident. Another person knowing your password can really get you into trouble. Similarly, if your password is compromised or you change the permissions of your files so that others can read them, they might copy your work and you might get caught when we compare submissions.
2. Ignorance of the `quota -v` command to monitor your network drive disk quota and the steps you must take to reduce disk space usage are likely to result in you account becoming unusable when you need it most. If this happens, corrective actions might take several days. Lateness of programming projects or lab exercises will not be excused, nor can any “urgent” system administration actions be taken.

F. Make-up Exams: Make-up exams will be given only for valid and verifiable excuses (e.g. a major medical situation). In such a case, it is your responsibility to contact the instructor ahead of time if at all possible.

G. Policy on I grades: A grade of I will only be given for genuine extenuating circumstances that are beyond your control after the midterm point. Both of the following conditions must be met:

1. Your work must be in good standing as of March 16, 2011 (two weeks before the drop date), defined as follows: You must have an average score of at least 50% on the programming assignments and at least 50% on the lab exercises due up to that point; and further, your midterm grade must also be equivalent to at least a C. Therefore, if you miss the midterm or have performed poorly on programming assignments or lab exercises, you are not eligible for an I grade.

2. Written documentation must, upon request, be supplied about the extenuating circumstance either by you or by the University administration. The Dean of Undergraduate Studies and her assistants are there to assist you and will write letters to your professors that request appropriate accommodations.

Under no circumstances will the condition for completing an I grade be that the entire course be retaken later without a new registration.

H. Disabilities, etc: Accomodations will be made for clients of the Disability Resource Center upon adequate prior notice and according to that center’s policies.

Students with genuine continuing hardship situations, or any disability related problems with Lab or homework computer usage should confer with the professor before February.

I. Attendance: You are responsible for all material presented in the lectures. Some of that material will not be presented anywhere else.

Sometimes quizzes will be given in lectures. They will be counted into your grade. No makeups will be given. The zero score will be excluded from score average in case of genuine, documented extenuating circumstances.

Make sure you have a trusted friend to lend you notes about what the professor said or emphasized, and the content of class discussion if you are going to miss a lecture. However, materials shown or distributed will be posted on the course web site shortly after each lecture.

J. Other Notes:

1. During their office hours, the instructor and the teaching assistants for this class will be glad to help you with the course material and the programs.

2. In addition to the regular office hours, you can also set up an appointment to meet with your instructor and the teaching assistants.
3. We will answer “reasonable” questions by email; which means definite questions that have definite and short answers; and which are at the academic level of the course and on course topics. We may edit and post on the Web questions and answers we think would be helpful to others (but I will email you a personal reply too.) We will remove your name to preserve privacy, unless you specifically state in your message that you would like your name included if we decide to post the question. We might ignore any others. We will not write or copy long explanations that repeat material we presented or are available for you to read.