CSI 310 – Data Structures – Spring 2012
http://www.cs.albany.edu/~sdc/CSI310 (Official location for announcements)

Course Policies

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

Prerequisites:  ICSI201 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 assessment will be administered near 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 will be reused, so students whose CS1 course did not cover these are expected to take special effort to catch up.  All the information you need of that is in CSI310’s assigned textbook, so you should not have to buy the introductory multimedia computing book if you don’t have a copy already.

Many find this course fast paced and challenging.  But the multimedia components make it more fun and make imagining what’s in the computer easier.  The conceptual difference between a data value, like what color a spot on an image is (represented by three numbers, for red, green and blue) and the location of the spot (represented by x and y coordinates) is generalized to Java objects and references (pointers), and is a major course topic.

CSI310 is the second in a two-course sequence for computer science majors and minors.  It’s purpose is to enable you to (A) work with and/or (B) begin to become one of those expert people who invent or implement software and maybe, as research computer scientists, develop ideas about computing that fit precise observations or apply to hard problems better and better.  People who invent or implement software are today called software developers, software engineers or architects, sometimes system or other “analysts”; sometimes they are still called computer programmers.  This course will also help you to become able to write substantive programs or scripts for research computations, customizing many of today’s professional software products, or web sites using technology like JavaScript.

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 1\(\frac{1}{2}\) weeks between taking this course and CSI201.  This semester I also teach CSI201 in LC-18 on Tuesday and Thursday, 4:15PM-5:35PM.  I welcome anyone to attend the CSI201 lectures, CSI201 labs if space is available, and I will facilitate students switching their registrations between CSI201 and CSI310.

Required Materials:

1. “iClickers” will be used for frequent in-lecture preparatory reading quizzes, warm-up exercises, thinking exercises, skill and concept quizzes, and surveys.  Which iClicker questions count for correctness and which just count for participation will be announced.  You MUST get, register, bring and use your iClicker in the lectures on Tuesday, Jan. 24 and/or Thursday Jan. 26 to AVOID A grade-affecting penalty.  However, everyone will be excused from the clicker components of up to 3 missed lectures, ON or AFTER Jan. 31.
If you don’t have an iClicker from last semester, BUY ONE NOW from the UAlbany or other college bookstores, or used from another student. And everyone get a supply of fresh batteries!

To get iClicker grade points, you must register your iClicker with your name, UA NetId and the code printed on your iClicker’s back by using the web site http://www.iclicker.com/registration. Doing this before the next lecture, when iClicker work begins to count, will ensure you getting all the grade credit. However, if you registered your iClicker for a UAlbany course this past Fall, such as in Ekblaw’s ICSI201, you need not register it again. (iClicker registrations expire each June.)


A guide for the first two week’s reading is on the web site: Not having the book is no excuse for these reading assignments.

Java libraries (“Book Classes”) and entire code of the example programs supporting the text. We will continue from those authors’ introductory book, but you must download and use the “Media Classes” for their data structures book for this course; NOT your copy of the “Book classes” from CSI201.

Reading and “warm-up” assignments will be assigned for which clicker QUIZZES and warm-up exercises will be given in the very next lecture. I expect every student will at least have access to the textbook. The first 4 chapters are posted for free access at http://coweb.cc.gatech.edu/mediaComp-teach#DS (to give you time to decide whether to stay in the course and to buy the textbook.) Use of the book will not be allowed during exams, so classmates sharing a copy is feasible.

Programming Project Zero and Reading), DUE Monday, 12:00 Noon. 80% if received after that but before Friday, 5:00PM.

Read this syllabus to prepare for a syllabus quiz!

Programming: 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, and/or go to lab tomorrow (Fri, Jan 20) afternoon. If you want to, make the program more interesting but limit it one class and to 20 lines of code.

The first lecture will show how to run DrJava on UA “Information Commons” Windows PCs. It’s critical to right away use the software mechanics to save your work in .java files and submit them on Blackboard for grading and credit, so this won’t be a distraction when you do harder assignments for homework and labs.

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

5. Optional, recommended textbook: Gaddis and Muganda, “Starting Out with Java, from Control Structures through Data Structures”, any edition, Addison/Wesley/Pearson. This book explains and gives examples of Java features and data structures from the beginning. It is a good place to read, collected in one place, what you need to know to do a particular thing, like make and use an array, your own class, some data structure, etc.

6. Course lecture notes, supplemental readings and external Web references provided during the course will be posted on the course Web site.

7. Your CS1 Java textbook and/or other brief references for Java programming fundamentals.

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.

   Briefly: Given a sufficiently detailed description of what a computer can do and a strategy for doing it, the student shall write a program that makes the computer do it by implementing the given strategy.

2. Solve programming and (paper) analytical problems with solutions, answers and explanations that demonstrate principles, practices and elementary analyses of some specific data structures and program control structures.

   These (more or less non-obvious) data structures are well-known and understood by computer scientists, professional programmers and software designers. Most of these data structure involve or actually contain data that refers to data, besides or instead of data that is meaning to people. Data locating or referring to data produces the power, efficiency and, for many beginners, the learning difficulty of these linked and/or arrayed data structures.

   Within these data structures, there are pointer or reference and/or array index DATA whose purpose is 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.

   The control structure topics in this course beyond those introduced in ICSI201 are recursion and details of how method calling and returning is related to the local variables of methods which are managed in the stack of activation records. Method call and return is a powerful problem solving, algorithm structuring and computer coding technique. Method calls are recursive when the code in a method’s body directly or indirectly calls the same method. Conceptually, recursion means to define something in terms of itself in a way that avoids the philosophical problem of circularity. In ICSI310, you will write recursive methods and
demonstrate how they work by diagramming the stack of activation records which hold local variables and by diagramming the entire computation by drawing a tree. (By the way, abstract mathematical approaches to recursion are taught to computer science majors worldwide; Albany’s course for this is ICSI210.)

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. Similar projects and problems will cover arrays, linked lists, stacks, queues and networks.

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: The schedule for lab sessions and staff office hours will be linked from the course web page and Blackboard.

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 or conflicting, we will accommodate 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 space available (which there is likely to be.)

iClicker and paper exercises and quizzes will be given in lectures to monitor student preparation, progress, learning needs and attendance. Points for correctness and participation will be included in the grade.

2 Lab meetings will begin Monday (1/23).

But, we will hold a drop-in orientation and catch-up session this Friday 1/20, 1:00PM-5:00PM in HU-25, particularly for transfer students. 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>Assessment</th>
<th>Date/Time</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterm</td>
<td>Mar. 8, 2012 (Thurs)</td>
<td>18%</td>
</tr>
<tr>
<td>Final</td>
<td>May 15, 2011 (Tues)</td>
<td>27%</td>
</tr>
<tr>
<td>iClicker Score</td>
<td></td>
<td>13%</td>
</tr>
<tr>
<td>Lab. Exer.,</td>
<td></td>
<td>12%</td>
</tr>
<tr>
<td>Homework Projects</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 cutoffs will be determined by the quality of fundamental content the scores represent, so there is no predetermined percentage A’s, B’s, C’s, etc. to be awarded. If most people fully master the fundamentals, most will get A’s!

The cutoffs will also be determined 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 ten to fifteen programming project assignments. Some will be continuations of previous ones.

Submissions that do not compile get ZERO points, automatically! Incremental software development will be demonstrated in lectures, with student iClicker polls for planning and design decisions; outlines for it will be included in some assignments. 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 after getting those error messages from the Java compiler.

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. Information Commons computers are provided in the UA Libraries as the “last resort” for students with computer or network problems.

HOWEVER, delays will be granted in case of weather problems or problems with the Blackboard or Information Commons UAlbany infrastructure.

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

D. Policy on Cheating:

1. Most assignments, iClicker questions and all examinations in this course are individual exercises. The work that you do must be yours—not that of other students, friends, tutors, Web volunteers or posters, etc.

While it may seem like the easy way out and a time-saver for doing assignments to copy what you’re assigned to figure out yourself, this strategy will backfire on tests. It will backfire because we design the tests for you to demonstrate specific thinking and problem solving skills. Those skills are developed by doing assignment work and other practice, like strength
in muscles, in your own individual mind, stored ready to use the more quickly in the brain inside your head by the more neural connections made by personal practice (and sometimes puzzlement.) It will again backfire when discover yourself unable to do subsequent assignments that rely on the skills you missed developing because you copied the work of someone else who actually has those skills. Finally, it will backfire when a job interviewer asks you to solve a data structures problem on the spot and explain your reasoning, or simply asks “I see you took data structures at Albany. Was that the CSI310 class on your transcript? Can you demonstrate for me on the whiteboard when and why linked lists are faster than arrays, and vise versa?” Your “friend” who lent you his or her excellent CSI310 solutions will get the job, not you!

It will probably backfire even if it feels like you understood the solution at the moment when you copied it! This happens for the same reason that your body cannot learn a skill like ice skating or music playing by just watching. For the same reason, many people who follow GPS directions when they drive just don’t remember the route when the GPS is not working! Making you mind struggle with programming puzzles actually builds physical connections among the neurons in your brain. Doing that repeatedly makes you do it faster and more confidently each time because it strengthens the newly made connections.

You are encouraged to form study groups, discuss assignments and techniques in general terms, etc., but the assignments themselves must be your own work. In particular, two or more people may not create an assignment together and submit it for credit.

The particular rule is that you are encouraged to explain to each other anything for the course in English (or other natural, human language like Chinese or French) and in diagrams, geometric sketches, etc.

BUT, you must not communicate by typing or dictating, nor copy (from people or sources other than the textbook and this course’s notes) by pasting or manually typing, actual Java code that constitutes parts of solutions to assignment problems. Nor may you communicate text-editing instructions that may be followed without thinking or understanding.

You might find or invent fragments or examples of Java code that may be helpful for others to see, but which will not cause classmates to fail to learn because enables them to skip thinking for themselves. If there is any doubt, you must pass them through the professor (by email) to review before posting them on discussions or otherwise communicating them.

For example, if you show a classmate or tutor code with bugs, and he or she notices that a variable is not initialized properly, he or she should tell you something like “the variable you seemed to use for counting something was not initialized properly”, but NOT “Just change the 0 in your line ‘int count = 0;’ to a 1 and you should be fine.”

2. Academic Integrity and iClickers: iClickers are frequently used for individuals to take quizzes in class on what they should have studied, paid attention to and practiced. It is a serious instance of cheating to use another student’s iClicker in class. This includes multiple iClickers, etc. (These things are called “fraud”.) Our 3 missed lecture policy is intended for you keep up your course grade even if you choose to miss class occasionally. Any instance of cheating with clickers is serious, and handled as other incidents of academic dishonesty as follows:

(a) A zero score, for all students involved for the affected assignment, lab or the entire lecture (zero averaged in, not a missed lecture) where an iClicker incident occurs.
(b) A written report will be sent to the Vice-Provost for Undergraduate Studies, according to University policy intended to detect cheating in a multiplicity of classes.

(c) For repeated cheating or cheating on an exam by impersonation or communication, a failing grade and/or a referral of the matter to the University Judicial System for all students involved.

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 or other University 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, grade reduction or academic integrity reports like for cheating can be imposed by the professor 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 set the permissions of your files so that others can read them, they might copy your work.

2. Learn and use the `quota -v` command to monitor your network drive disk quota and the steps you must take to reduce disk space usage. Reaching the limit is likely cause you to lose your latest work at a time when you are most busy! Lateness of or loss of programming projects or lab exercises will not be excused for this reason. Retrieving lost work from ITS backups takes days at best and might not be successful.

   You can avoid unnecessary use of your quota by using CSI310 shared copies of Guzdial and Ericson’s media files instead of keeping the download of your own copy. The ITS Unix pathname (location) will be announced when needed.

**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 21, 2012 (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 everything else 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 exams, assignments, etc., you are not eligible for an I grade. Your
choices are to work smartly and hard to catch up, get a poor or failing grade, or drop the
course by the drop date of Wed. April 4, 2012.

2. Written documentation must, upon request, be supplied about the extenuating circumstance
either by you or by the University administration. The Vice-Provost for 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 repeating the entire course
without a new registration.

H. Disabilities, etc: Accommodations will be made for clients of the Disability Resource Center
(DRC) upon adequate prior notice and according to that center’s recommendations. Many Albany
students had much greater educational success after strategies to work around a wide range of
mental and/or physical personal variations were recommended by the experts at UA’s DRC. Of
course privacy prevents their identities from being disclosed.

Students with genuine continuing hardship situations, or any disability related problems with
Lab or homework computer usage should confer with the professor (and DRC for disabilities) before
February.

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

iClicker and/or paper quizzes, polls and learning exercises will be done in lectures. They will be
counted into your grade, but up to three will be excused automatically. No makeups will be given
because of their participatory nature. Comparable but unscored material and limited tutoring
will be offered for missed activities. In cases of genuine, documentable extenuating circumstances,
additional activities will be excluded from score average; and assignment deadlines may be delayed.

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

J. Other Resource Notes: During their office hours, the professor and the teaching assistants
for this class will be glad to help you with the course material and the programs. They will ask
you about your having read the assignment. You can also set up appointments. Or drop in on the
professor when he’s not too busy.

We (prof. or TAs) 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 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.