Chief learning goal: The ability to demonstrate in the written forms of English, calculations, graphs, pseudo-code and Java code valid applications of algorithmic thinking to somewhat novel procedures and computational problems.

Other learning goals: (a) Implement, test, debug and measure performance on big and small problem sizes of algorithms based on their exposition and pseudo-code descriptions in textbooks like CLRS. (b) Write reports of what are the topics and facts stated and/or proved in specified sections or chapters of textbooks like CLRS. (c) Write demonstrations on paper of algorithmic ideas applied to novel problems and runs of algorithms described in English, pseudo-code or practical code.

Homework will consist of three kinds of activities: Theoretical questions, including simulation of algorithms on paper; programming of algorithms presented or variations thereof; and experiments to directly observe and quantify the time a computer takes to run a program running an algorithm that you know. Aside from solving the problem at all, predicting the time consumed by the computer is the chief underlying practical problem that the whole subject addresses.

Lectures will include the same three parts combined with active learning activities: Quizzes, some to determine teaching needs and others to evaluate preparation and in-class learning, short responses collected and discussed in class, “think, pair and share” discussion, etc. In lieu of taking attendance, written responses will often be collected, evaluated and counted in the grade. Therefore, makeups for missed in-class activities will not be given. However, if you miss a class because of an genuine extenuating circumstance, that class’s zero scores will not be averaged into the average in-class score. (The instructor may require documentation for excuses to go beyond 3 absences.)

This is a standard subject in computer science. Its nature is explained in algorithms textbooks such as the one required for this course: Introduction to Algorithms, 3rd edition, Corman, Leiserson, Rivest and Stein, MIT Press 2009. We will cover a selection of topics all with the intellectual approach that works as follows:

1. We focus on various problems that are useful simplifications of practical problems, and express them in mathematical language so the problems are absolutely well-defined.

2. We work out how one or more fundamental algorithmic ideas are developed and combined to apply to each well-defined problem. The resulting combination, in this course, is also related to experience with its implementation in a real computer.

3. We often identify and compare how different algorithmic ideas or combinations apply to the same problem. Separating the problem from the single algorithmic solution someone had learned first enables an expert to identify, evaluate, and perhaps choose the best among multiple alternative solutions.

Grades are based on cutoffs applied to a score calculated with the following weights: Homework: 25%, Midterm exam (March 17 in class): 20%, In-class score: 20%, Final exam (May 12, 1:00PM-3:00PM): 35%. However, I reserve the right to award a higher grade based on substantial improvement between the times of the midterm and the final. The A-B cutoff will be roughly 80% and the B-C cutoff roughly 70%.

Each generally weekly homework assignment will consist of a up to six or so problems, and is generally due on Thursday. However, each individual problem will be accepted if it is up to 5 days late, with penalties: -5% for 1 day, -10% for 2-3 days, -15% for 4 days and -20% for 5 days. (Submit late paper work to the CS Dept. office.)

Instructor: Assoc. Prof. S. Chaiken. Office hours: LI-96H (Computer Science Dept. Area) T, Th 2:40PM-4:15; and Wed 9:30AM-11:00; also by appointment or drop-in when I’m not too busy. Calling 442-4282 will reach my office but I much prefer email messages to sdc@cs.albany.edu to voice-mail, which I might lose.
• Catalog description: CSI 403 Algorithms and Data Structures (3 credits) Description of common data structures such as lists, push-down stores, queues, trees, and graphs. Definition of algorithm efficiency and efficient algorithms for integer and polynomial arithmetic, sorting, set manipulation, shortest paths, pattern matching, and Fourier transforms. Prerequisite(s): CSI 210 and 310. Normally offered spring semester only. [Note: The particular topics may vary from this list but they are in the same spirit.]

• This course is required for UA’s two Computer Science Bachelor’s of Science programs.

• It an elective for the CS Bachelor’s of Arts and the CS minor. Students in all majors are welcome to take the course if they have the background, say from studies in Mathematics. It is one of the more challenging and theoretical courses among the CS electives. Knowledge of both prerequisite subjects is necessary. The prerequisites are may be enforced if the student does not demonstrate the prerequisite knowledge in the course. The prerequisite of CSI310 implies the ability complete to programming assignments and be familiar with basic data structures.

• Class meets T, Th 10:15AM-11:35 in BA-130.

• Use of computers for Web access, Blackboard and programming in Java is required. Computer Science students are encouraged to get their own computers for programming and experimenting with things such as multiple operating systems. However, the Information Commons PC’s in the libraries will suffice for meeting course requirements.

• Academic Integrity:

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 new code to be written and any written answers for every assignment and exercise 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 assigned code and answers he or she submits independently, without copying.

3. Cheating in an individual homework problem will result in a ZERO for that requirement for all the students involved. Students who cheat in two or more problems will receive an E grade for the course. A report of 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 Bulletins.