

CSE 392 – Programming Challenges

Spring 2012

At its best, computer science is an exciting blend of programming, mathematics, and problem solving. This course will introduce an interesting variety of subjects in programming, algorithms, and discrete mathematics through puzzles and problems which have appeared in the International ACM Programming Contest and similar venues.

Instructor: Steven Skiena

Office: 1417 Computer Science

Phone: (631)-632-9026

Email: skiena@cs.sunysb.edu

Office Hours: 2:15-3:45PM and by appointment.

Course Time: Tuesday-Thursday 3:50PM-5:10PM **Place:** ESS 069

Textbook: S. Skiena and M. Revilla, *Programming Challenges: The Programming Contest Training Manual* Springer-Verlag, 2003.

Recommended Books:

S. Skiena, *The Algorithm Design Manual*, second edition, Springer-Verlag, 2008.

R. Sedgewick, *Algorithms in C*

J. O'Rourke, *Computational Geometry in C*

B. Kernighan and D. Ritchie, *The C Programming Language*

- **Grading:** Grades will be assigned based on performance on the 14 weekly programming assignments ($4 \times 14 = 56\%$), the 100 problem review (20%), and class participation (24%).

Historically, students at Stony Brook who regularly complete at least one problem per week, do the review, and attend/participate regularly will likely get some form of B, while students who regularly complete at least two problems per week, do the review, and attend/participate regularly will likely get some form of A. My students in Hong Kong in Spring 2009 did much better, with several completing all 56 problems, so maybe we can raise standards here this year as well.

- **Homeworks:** There will be 14 problem sets over the course of the semester, one per week. The problems are due one week after being assigned, and you should not plan on being able to catch up on old problems in subsequent weeks. Each problem set will consist of 4 programming problems. The goal is to maximize the number of problems solved *correctly* (according to the robot judge).

- **The 100 Problem Review:** Each student will be assigned a block of 100 problems from the UVa judge and asked to read each problem and prepare a concise one line description of how interesting/hard it is and what techniques are used to solve it. I will provide more detail a month or so into the semester.
- **Exams:** There will be no midterms and no final exam. You will have an opportunity at the end of classes to describe the effort you put into the course and the results. Maybe we will run one or two timed programming contests for fun, perhaps with three-person teams to simulate the ACM ICPC tournament.

Rules of the Game:

1. The WWW page for the course is <http://www.cs.sunysb.edu/~skiena/392>. All course hand-outs and notes are available there, along with the latest announcements. Please check it out.
2. This semester's course will use my book *Programming Challenges* and the judging website www.programming-challenges.com. I will be very interested in your feedback on both matters.
3. Register for my (skiena-hk) classroom (Stony Brook – Spring 2012) with ID 1209 on the Programming Challenges Judge so I can monitor your performance. Note that there are several problems which will be assigned that only grade correctly on the UVa judge: these have been omitted from the PC classroom.
4. This course is designed to have a weekly flow. Each week, a new set of problems will be assigned, and the two lectures devoted to the algorithmic theory and ideas associated with them. Students should read the relevant problems before class so we can intelligently discuss them, and try them before the deadline so we can identify possible difficulties for discussion.
5. This course is intended to be a positive, fun experience for a small, enthusiastic group of students. You are not expected to complete all problems, but you can impress me if you do!
6. I will be happy to provide more advanced problems, if the students demand and the level of the class warrants it.
7. The grading of class assignments will be done by the Programming Challenges robot judge at www.programming-challenges.com. Students are urged immediately to go to the site and register. As this judging site is still new, expect difficulties. In particular, be sure you save copies of all your submitted programs, especially the working ones.
8. In the event that the Programming Challenges robot judge is broken, problems should be submitted to the Univ. de Valladolid robot judge, <http://online-judge.uva.es>. The actual judging scripts should be identical for both sites. Students are urged immediately to go the site and register as a new member (use your real name). Be prepared to email me your Valladolid name and ID number if we make extensive use of it so I can monitor your progress.

9. You are encouraged to do your programming in C language or C++, although Pascal and Java are also acceptable, although some people have experienced difficulty with using Java on the judge.
10. Students are to work on all problems *individually*. You may discuss the problems among yourselves, but you must write all your own code.
11. You are allowed to look in any books you wish, but are honor bound not to search the WWW for help. This is to mirror the official contest conditions.
12. There may be updates or clarifications to some problems on the Valladolid site, in either the HTML problem version or the newsgroup. You are encouraged to consult these sources.
13. The set of all class solutions will be submitted to similarity testing programs, to identify students who did not work on their own. *Since cheating in this course is a particularly stupid thing to do, I will be particularly energetic in pursuing any suspects.*
14. I expect my lectures each week will be somewhat informal. Full lecture notes are available on the course webpage.
15. My lectures from Spring 2009 were video recorded and have been made available at:
<http://www.algorithm.cs.sunysb.edu/programmingchallenges/>
This defeats the point of participating in a small, interactive class, but may be helpful on occasion. Older lecture notes, with audio are still available on the web, although the videos should be better.
16. The Pass/No Credit (P/NC) option is not available for this course.
17. If you have a physical, psychological, medical or learning disability that may impact your course work, please contact Disability Support Services office, 128 ECC Building (631) 632-6748. They will review your concerns and determine, with you, what accommodations are necessary and appropriate. All information and documentation of disability is confidential.
Students who require assistance during emergency evacuation are encouraged to discuss their needs with their professors and Disability Support Services. For procedures and information go to the following web site: <http://www.ehs.sunysb.edu> and search Fire Safety and Evacuation and Disabilities.
18. Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. Any suspected instance of academic dishonesty will be reported to the Academic Judiciary. For more comprehensive information on academic integrity, including categories of academic dishonesty, please refer to the academic judiciary website at:
<http://www.stonybrook.edu/uaa/academicjudiciary/>
Adopted by the Undergraduate Council September 12, 2006

Lecture Schedule:

1/24 arrays and iteration problems out
1/26 programming style / understanding specifications

1/31 data structure problems out
2/2 elementary data structures

2/7 string problems out
2/9 strings and library functions

2/14 sorting problems out
2/16 sorting and library functions

2/21 arithmetic problems out
2/23 arithmetic and algebra

2/28 combinatorics problems out
3/1 recurrence relations and counting

3/6 number theory problems out
3/8 divisibility and modular arithmetic

3/13 backtracking problems out
3/15 constructing subsets and permutations

3/20 graph traversal problems out
3/22 DFS/BFS

3/27 graph algorithms problems out
3/29 shortest paths and MST

4/2-4/6 Mid-Semester Break

4/10 dynamic programming problems out
4/12 edit distance and applications

4/17 grid problems out
*4/19 rectangular and hexagonal grids

*4/24 geometry problems out
4/26 geometric primitives and trigonometry

5/1 computational geometry out
5/3 convex hulls and triangulation

(*) implies there will likely be a substitute instructor that class.