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:** 3538 Academic Building (lifts 25/26)  
**Phone:** (852)-2358-6972  
**Email:** skiena@cse.ust.hk  
**Office Hours:** after class and by appointment.

**Course Time:** Monday-Wednesday 10:30AM-12PM  
**Place:** 2302 Academic Bldg.  
**Teaching Assistant:** Zhang Bo  
**Email:** zhangbo@cse.ust.hk

**Recommended Books:**  
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 × 14 = 56%), the 100 problem review (20%), and class participation (24%). At Stony Brook, students 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.

**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. As a visiting professor here at HKUST, I am unfamiliar with local rules, customs, and practices. Please inform me of any mistakes I make, as they are due to ignorance rather than malice. Corrections and changes to course policy will be made as appropriate.

2. The WWW page for the course is http://www.cse.ust/~skiena/300. All course handouts and notes are available there, along with the latest announcements. Please check it out.

3. The lectures will be video recorded by the Publishing Technology Center (PTC) and made available on the web. If you have any deep seated objections to being recorded, let me know ASAP.

4. 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.

5. 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 relevent problems before class so we can intelligently discuss them, and try them before the deadline so we can identify possible difficulties for discussion.

6. 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!

7. I will be happy to provide more advanced problems, if the students demand and the level of the class warrants it.

8. 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.

9. 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.
10. 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.

11. Students are to work on all problems individually. You make discuss the problems among yourselves, but you must write all your own code.

12. 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.

13. 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.

14. 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.

15. I expect my lectures each week will be somewhat informal. Full lecture notes from last year, with audio are available on the web.

16. The Pass/No Credit (P/NC) option is not available for this course.

17. Because a primary goal of the course is to teach professionalism, any academic dishonesty will be viewed as evidence that this goal has not been achieved, and will be grounded for receiving a grade of F. (See CEAS Procedures and Guideline Governing Academic Dishonesty, 1/81.)

18. If you have a physical, psychological, medical or learning disability that may impact on your ability to carry out assigned course work, I would urge that you contact the staff in the Disabled Student Services office (DSS), Room 133, Humanities, 632-6748v/TDD. DSS will review your concerns and determine with you what accommodations are necessary and appropriate. All information and documentation of disability are confidential.
Lecture Schedule:

2/2 arrays and iteration problems out
2/4 programming style / understanding specifications

2/9 data structure problems out
2/11 elementary data structures

2/16 string problems out
2/18 strings and library functions

2/23 sorting problems out
2/25 sorting and library functions

3/2 arithmetic problems out
3/4 arithmetic and algebra

3/9 combinatorics problems out
3/11 recurrence relations and counting

3/16 number theory problems out
3/18 divisibility and modular arithmetic

3/23 backtracking problems out
3/25 constructing subsets and permutations

3/30 graph traversal problems out
4/1 DFS/BFS

4/6 graph algorithms problems out
4/8 shortest paths and MST

4/9-15 Mid-Semester Break

4/20* dynamic programming problems out
4/22* edit distance and applications

4/27 grid problems out
4/29 rectangular and hexagonal grids

5/4 geometry problems out
5/6 geometric primitives and trigonometry

5/11 computational geometry out
5/13 convex hulls and triangulation

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