CPSC 202: Programming and Problem-Solving Practicum

Successful computer problem solving relies not only on the development of appropriate algorithms, but also on the recognition of instances of well-defined problem classes for which algorithms already exist. In addition, excellence in computer programming cannot be developed without significant practice in implementing and debugging solutions. In this course, you will program solutions to selected problems that will exemplify the various problem classes. At least part of every class period (except possibly the last day) will be spent programming. Class will also include discussions of problem-solving strategies and solutions.

Professor: Ellen Walker  
Email: walkerel  
Office: 112 Colton, x 5250

Lab Assistants: none, unfortunately

Meeting Times and Locations:
MTRF, 10-1, Colton 106

Web Sites:
http://cs.hiram.edu/~walkerel/cs202 General course web site.

http://acm.uva.es/problemset  
ACM Problem Set Archive with Online Judge

http://www.programmingchallenges.com  
Textbook web site – includes example programs!

Grading Scheme:

Your grade will depend on four components

Participation  10%
Judged Problems  40%
Notebook  30%
Original Problem  20%

There will be no exams for this course; only programming assignments.

Participation includes class attendance, participation in class discussions, and in-class exercises as assigned. It also includes your participation in the final programming contest (for which the winners will receive extra points). Each student is expected to present one of their solutions to a problem (either initial or final) sometime throughout the course for full participation credit.

Judged problems are problems that have been judged to be correct by the online judge at the Programming Challenges website (above). All decisions of the online judge are final. Problems that we have gone over in class will not count. You will be graded strictly on the number of problems solved: 1 problem = D, 2 problems = C-, 3 problems = B-, 4 problems = A-, 5 problems = A. Each judged problem must come from a different chapter of the textbook. For a B- or above, at least 1 program must be level 2 or above. For an A or A+ at least 2 programs must be level 2 or above. All problems must be judged by 12:01am, Tuesday, December 20, 2005.

Your notebook will serve as a record of each problem that you have attempted. These will include problems done as class exercises as well as judged problems. For full credit, your notebook must include at least 8 problems. For in-class problems done as teams, each team member can have a copy of the same
information in their notebook, with all names on it. Your notebook must be submitted to me by 5pm, Tuesday, December 20, 2005. Even though you are allowed to continue working on judged problems past this time, you should submit your journal on time. If you complete a problem after submitting the journal you can email an update (final code and comments) by midnight.

For each problem, the notebook should contain:

Identification of the problem.

Initial analysis of the problem, including problem type, ideas about how to get started, major data structures to use, etc.

First-draft pseudocode of the solution

Final (or last attempt at) solution

A record ("journal") of your progress as you worked on the problem – what issues came up and how did you resolve them? What did you learn as you worked on the problem? Include printouts of significant steps in the solution process (e.g. first syntactically correct version; first version that gets at least some test cases correct; one version for each major revision). This record should include the time spent on each program, including 'time stamps' (i.e. how much time has been spent on the program so far) for each printout.

Summary and evaluation of the solution: What general techniques or code from this solution can carry forward to future solutions? Did you actually solve the problem? If not, for what cases does it work, and for what cases does it not? If you believe that the solution is correct, but it is not being judged correct, write that here as well.

The programming contest will be held on the last day of class, Tuesday, December 20. You will program in teams (of 2 or 3 students), one computer per team. All students must participate in the contest. You will be given a problem set consisting of your original problems (don't give these out before the contest!) Your grade from the contest will be determined by the number of problems that you solve (excluding your own of course!) The winners of the contest (most problems; ties broken by shortest time and fewest resubmissions) will receive prizes, as well as acclamation.

Each contest team will submit an original problem by Monday, December 19 at 1pm. These problems will be used in the programming contest. Each submission must include a problem statement, a working solution to the problem (to avoid impossible problems!), a set of test data with correct answers, and a text description of the problem's classification and a suggested solution technique. Any "tricks" to the problem should also be described as part of the submission. Original problems will be graded on the basis of clarity, originality, "realism" of the description, how well they exemplify the stated problem class, and how well the test data covers all possible cases. The submitted program must work perfectly on all submitted test cases. (Clarity and adequacy of test cases will be evaluated during the programming contest!)

Honesty Policy:

The purpose of this course is to improve your programming. The only way to improve your programming is to program (yourself). Therefore, I expect any program submitted to the judge or included in your notebook to be your own work, unless specifically noted as a team problem (and including the names of all team members). It is acceptable to discuss general algorithms or very detailed debugging (e.g. resolving a syntax error) with each other, but all such help must be reported in your journal. Any evidence of programming that is not your own will result in a 0 on that assignment. Assignments judged to be dishonest cannot be made up.
Tentative Schedule

12/1: Chapter 1, Section 2.7

12/2: Chapter 2, Chapter 3

12/5: Pair Programming

12/6: Chapter 4

12/8: Chapter 6

12/9: Chapter 7

12/12: Chapters 9, 10

12/13: Review / Catch up

12/15: Chapters 8, 11

12/16: Chapter 13

12/19: Discuss 2005 Programming Contest Problems

12/20: Notebooks due, Programming Contest