

Course Proposal: Problem Solving in Computer Science

Frank Chu

July 15, 2004

Course Content

We would like to propose a Computer Science Problem Solving Course. Problem solving is an important part of Computer Science, and we feel that it deserves more attention than most CPSC courses offered at UBC give it. Many of the courses have lengthy assignments and complicated projects. The beauty of programming problems is their brevity and the ability to focus on a particular idea or technique without the need to keep in mind the details that commercial programming is full of.

A problem solving course has been offered in the past in the department of mathematics (MATH331), and it has been very successful. Our course will focus on a number of important programming techniques and concepts (dynamic programming, graph algorithms, number theoretic algorithms, computational geometry) with problems that illustrate their applications. Some of those concepts are taught in upper level algorithms courses (CPSC320 and CPSC420), but those courses are mostly theoretical. Students do not get a chance to try and implement the algorithm themselves.

There are several groups of students who will find this course interesting. First, there are second- and third- year students who want to see interesting problems and programming techniques which they would not otherwise see until fourth year. Second, there are those who are familiar with the theory but would like to see its applications without having to create large and detailed software projects. And finally, there are those who would like to participate in the annual Association for Computing Machinery (ACM) Collegiate Programming Contest, which is where many of our programming problems will come from.

The requirements for taking this course are as follows: CPSC216 (old curriculum) or CPSC221. However, students with sufficient familiarity with a programming language (C++ or Java) are also welcome. We intend to introduce all algorithms from scratch. We will focus on the practical implementation, application, and usage of these algorithms, and will leave the more abstract theoretical aspects to courses such as CPSC320.

Course Structure/Format

The course would be offered in the format of a seminar. The group would meet 2 or 3 times a week. Students would first present their solutions to last week's problems. Everyone would have a chance to present a problem and/or its solution (assuming it illustrates the current topic of discussion). Then a new topic would be explained and new problems on that topic would be presented to the students. The coordinators' tasks would be to explain the topic and illustrate it with previously prepared problems with working solutions. The coordinators would also assign homework problems, judge the effectiveness of students' solutions and offer improvements.

A CPSC graduate student and a previous participant of the ACM Programming Contest has agreed to be a frequent guest lecturer and would gladly attend most of the seminars.

Course Requirements and Evaluation

Each seminar will cover a particular topic, and students would be asked to complete weekly homework problems which would require the use of the particular concept or technique discussed. They would submit their solutions to a particular problem via an online judge system that we designed. The automated online judge system would receive code from students, compile and execute the program, and reply appropriately. The benefit is that students get immediate results to their submissions, and can then review their code and attempt to correct mistakes. This allows us to fairly assess students' ability to understand the concept of technique involved, while at the same time encourage them to implement the algorithms correctly. Solutions would not only be evaluated based on correctness, but also on conciseness and elegance.

Some problems can be challenging, so teamwork would be encouraged. However, the judge system also has plagiarism detection that attempts to maintain academic discipline. The coordinators will report all cases of plagiarism to the Faculty.

Participation would also be considered in the evaluation procedure. Written tests would feature theoretical questions as well as problems, but correctness of code would not be as essential as the use of the correct technique, because writing clean code on paper under time pressure is extremely difficult.

Potential Faculty Sponsors

We have at least three potential faculty sponsors in mind.

Alan J. Hu: Associate Professor, (604)822-6667, ajh@cs.ubc.ca

He was the mentor of last year's UBC ACM Collegiate Programming Contest team.

David G. Kirkpatrick: Professor, (604)822-4777, kirk@cs.ubc.ca

He was our instructor for CPSC320 and knows us personally.

George K. Tsiknis: Senior Instructor, (604)822-2930, tsiknis@cs.ubc.ca

We believe the department would give approval for this course.

Rationale for Why the Course Should Be Offered

There are no honours courses at the department of Computer Science, and this course would appeal to CS honours students. The course would fill a hole that we believe exists in the current CS curriculum. Most real life programming projects are collections of problems, so our course would be a very useful tool for building experience required to tackle real life projects without the bulkiness of a complex software system. The course would be very rigorous in presenting the topics and explaining the underlying mathematical theories, but the seminar format and small class size would allow for student interaction than one would expect in a lecture-based course. And of course, problem solving is exciting and challenging not only for computer scientists. Because of the low prerequisites students from other departments would be able to take the course.

We have also noticed that the SDS program has not offered many science courses in the past. We are very motivated and prepared to take on the challenge. We have over 1000 programming problems at our disposal (most of them with solutions) and a lot of problem solving experience. We feel confident that we can do a great job on this and are looking forward to designing and coordinating a seminar series.

Qualifications of the Coordinators

Frank Chu: 4th year Combined Honours in Computer Science and Mathematics

- * Student number: 79621017

- * Phone: (604)604-4397332

- * E-mail: fpmc@shaw.ca

- * Related experience: Undergraduate student facilitator for the Math 180 Workshop, 2003. UTA for CPSC 211, 2004. Twice a member of the UBC team for the ACM ICPC programming competitions.