Applied Computer Programming

Course Description
The Advent of Code is an international competition for computer programmers. The competition involves puzzles that are released daily for 25 days beginning on Dec 1 of each year. The puzzles are generally classic problems of computer science (for example: virtual assemblers, vectorization, memory-access bottlenecks, summed-area tables, satisfiability modulo theory, and graph traversal). Following approximately one week of preparatory work based on previous years' puzzles, the daily cadence of the contest shall begin. For the remainder of the block, each day students shall read the daily puzzle that is released to the world each evening, solve the puzzle for homework, and spend class time the following day discussing the solutions of their classmates and other published solutions from global competitors. Students must use at least two different programming languages during the block. Competitive ranking is not a means of assessment in the course.

Pre-requisites: IND 3156 or IND 3144

Course Objectives
The primary goal of this course is to be able to solve problems. In this course, the problems comprise fundamental logic and the manipulation of primitive and derived data types—every programming language is built upon these basics. A secondary goal of this course is for the student to be able to propose a solution to any computing problem in any machine language—proficiency in at least two languages shall be assessed. Another secondary goal is for the student to be able to disseminate and discuss code that is both readable and idiomatic—and to read and discuss others’ solutions to the same problems. Another secondary goal of the course is for the student to be able to identify classic problems and techniques as they manifest in the puzzles and solutions.

Students shall be required to obtain and use a github account in order to access the contest content. The student’s github identity shall be shared with the instructor. (The student may also optionally chose to share the identity with the class.) By maintaining a code base of puzzle solutions on github, many of the secondary goals of the course will be pursued.

Competitive standing in the contest is not a metric for success in the course. Rather, for example, many participants in past years have used the contest as an excuse to learn a new language rather than to “compete” in their “best” language. In this course, the primary focus will be an analysis of the problems and their solutions—including the choice of language and approach. Students must author their own solutions; those solutions shall be judged on the identification and application of fundamental techniques: not on the speed with which the student was able to input the code. Nevertheless, observing the culture and techniques of the international competitive community shall provide a valuable discussion piece in the course.
Assessment and Grading Information
The grade in this course is not curved; every student can get an A! The course grade shall be formed as follows:

- 60% progress on puzzle solutions
- 25% class participation in discussion and academic citizenship
- 15% preliminary report on previous contest(s)

Furthermore, the requirement to use at least two different programming languages during the course shall be enforced as follows. An overall scaling factor shall be applied to the score comprising the three preceding items. The scaling factor is determined from the following table:

<table>
<thead>
<tr>
<th>factor</th>
<th>criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>different puzzle answers are obtained in two (or more) languages</td>
</tr>
<tr>
<td>0.95</td>
<td>the same puzzle answer is obtained in two (or more) languages</td>
</tr>
<tr>
<td>0.90</td>
<td>one language is successfully used, other languages are attempted unsuccessfully</td>
</tr>
<tr>
<td>0.80</td>
<td>multiple languages are attempted but none are employed successfully</td>
</tr>
<tr>
<td>0.75</td>
<td>only one language is used and no answers to puzzles are successfully obtained</td>
</tr>
</tbody>
</table>

60% — Progress on puzzles shall be assessed by the instructor reading the students’ code and comments on github. Since solutions to the puzzles may be found on the internet within minutes of release, there is no marking value for correct program outputs in the absence of documented development of techniques. Indeed: readable, idiomatic, documented, published code shall be assessed using the attached rubric. Correct outputs need not be obtained to all problems if sufficient appreciation is demonstrated of the necessary techniques. Feedback on assignments shall come in the form of scores on the attached rubric and as generalized discussion in class.

25% — Class time shall be used to compare approaches and successes and failures, generally in a break-out-and-report-back format. Plugging away on the puzzles shall be a solitary endeavour outside of class—we owe it to each other to have lively interchanges during class! A commitment to github dissemination and to class participation is as important as the puzzle content.

15% — Since the Block shall begin before the contest content begins on Dec 1, the opening days of the course will involve preparation. There shall be class-wide exercises concerning the rubric, then each student shall prepare a report on some aspect of contest(s) past. This report may be

- the authoring of a well-prepared and published solution to a past puzzle;
- an analysis of an exiting published solution to a past puzzle;
- an analysis of the user-community response to a past puzzle or solution.

Other content is also permissible, with instructor permission. The report shall be submitted electronically or in hard copy to the instructor on a date before Dec 1 to be announced, but the preliminary report need not be published on github.

In the end, a letter grade (with finer pluses and minuses) is assigned to the average on the scale A (91–100%), B (81–90%), C (71–80%), D (60–70%), F (<60%).
Expectations and Policies

Assignments
2. Near-daily work on puzzles: of all of the in-block weeknights upon which a puzzle is released, all but two shall be assessed. Friday- and Saturday-night puzzles are not assigned but may be substituted for missed weeknight assignments, at the request of the student. In the rare case that the University excuses you from obligations (e.g., death in the family), your mark shall be formed as though the missed obligation(s) never happened.

Attendance
1. Please show up on time or not at all.
2. Absence from class results in a zero for participation that day. In the rare case that the University excuses you from obligations (e.g., death in the family), your mark shall be formed as though the missed obligation(s) never happened.

General
- drinks in the classroom are OK; food in the classroom is not OK
- do the homework, both reading and written work; I don't assign much
- make use of office hours or make appointments, if necessary

Exceptionality Accommodation
If you have an exceptionality for which you seek accommodation, please make sure to have registered with the Learning Commons, as specified in the Student Accommodation Policy, and provide me with your Memorandum on the first day of class. Beware that the contest has a daily cadence that waits for no one.

Communication
During the week, I generally check email periodically from 8 A.M. to 6 P.M. Do not count on receiving a timely, same-day response to emails sent after 7 P.M. on a week night or anytime on a weekend; please plan accordingly. Please remember to treat email to your tutors as a formal means of communication, one that demands courtesy and respect—use proper salutations, forms of address, punctuation, grammar, and syntax. Good writing begins with everyday practices.

Office Hours
I have an open-door policy and welcome you to join me in my office anytime that I am free. My posted hours during which I am available on a FIFO queue are Mon, Tues, Fri 10:00–noon. Beware that I often have faculty obligations noon–1 P.M. and 4:15 P.M.–closing (meaning both email and availability blackout). In general, you shall be expected to solve syntax problems on your own; office hours and class discussions (with the exception of dedicated break-outs) should be reserved for conceptual content.
Readings and Required Books
There is no textbook for the course and there is nothing to purchase. A course page shall be maintained at gitlab containing code examples and links to useful free internet resources. The course page will also include interpretive commentary by the instructor as the contest progresses. Course texts and other useful readings shall be made available via PDF on our moodle page or via hyperlink on the course page. Assignments and course emails will use moodle.questu.ca.

General-purpose online texts
• How to Design Programs, 2/e. Felleisen et al. https://htdp.org/

Scholarly articles that inform our problem-solving approach and rubric
• Mustapha et al. (2016). “Generic Assessment Rubrics for Computer Programming Courses.” TOJET 15, 53

Additional Information
Please bring your own questions, tangential or otherwise, into our discussions. I will do my best to make our discussions lively and exciting, but that task ultimately comes down to you. It is easy for all of us to slip into lecture/listen mode; let’s all make sure that doesn't happen.
Ethical Standards and Academic Dishonesty

Our course page has several links to ethics and dishonesty policies that we shall discuss in the opening days of the course.

Strong arguments can be made for recycling code that is known to work. In that sense, I shall not deny you the opportunity to cut-and-paste code for ancillary functionality like file I/O. More so than in an introductory course, in this course, my priority is for you to identify, apply, integrate, and modularize any necessary functionality, not necessarily for you to invent or keystroke it yourself.

As such, I am not strictly incorporating the common prohibition against sharing code. To be sure, sharing code can have consequences for you as a learner that you may not appreciate, but I am being very careful to distinguish instructional techniques from ethical mandates. I feel that the suggestion in the MIT policy to "make a habit of closing your laptop while you’re helping [a classmate]" is excellent on many levels.

In list form, the standards for this course are:

1. Any code acquired from a different source (whether from an official manual, an internet forum discussion, from an author's personal repository, etc.) must include the source URL or citation in a comment in the code.
2. The comment must appear at the location of the code in question and should delimit the beginning and end of the snippet, if necessary. Do not simply bury a list of references at the end of the file or in a separate Readme file. Anywhere that the borrowed code is used again by you should be accompanied by the same comment.
3. Git commit the file with the new code with an informative message like 'found some file I/O code'.

The last of these requirements is arguably going to slow you down because it will divert you from your paste-compile-test-from-all-of-my-open-browser-tabs groove. Too bad. When your program actually starts working, take a moment to paste the comment and commit the change—I have included the git directive on my list in order to encourage the development of habits that are not only ethical.

The line between ancillary functionality and essential creative input is something that we need to draw together in discussion. Especially concerning for our course—which is based on a famous contest with a large internet footprint—is that myriad answers and complete puzzle solutions can readily be found online within hours of release. I'm taking the position that requiring your work to be posted to github will prevent you from wholesale-pasting internet solutions—I'll shepherd your conscience; future employers can judge your desirability. My appealing to the honor of a thief is probably a fool's errand, but part of Quest's experiment is to see whether a student can be engaged enough to learn to pursue a problem for its own sake.

I cannot police thoughts; I must police actions. Although pasting source comments is a practice that I am trying to encourage, I have not included ethical behavior as part of the rubric. That is, the ethically-mandated comments that are described here are not part of your Documentation or Readability or Github scores. Rather, if the preceding practices are not followed, the offending assignment receives no credit and I'll follow up with the University as appropriate.