# CSCI 1312 (Introduction to Programming for Engineering), Fall 2017

## Review for Final Exam

## 1 Format of the exam

The exam will be at the scheduled time for the course final, December 12 at noon. The exam will be about twice the length of the first exam (hence intended to take about two hours), but you can use the full three-hour period if you like. Like the quizzes, it is "open book / open notes", which means you can consult paper or electronic copies of the textbook and your notes, sample solutions *from this year only*, your own graded work, and anything on the course Web site. You may not use other books, materials from this course from previous years, a calculator or computer (except as needed to consult allowed sources), or (of course) each other's papers. *Notice the restriction on computer use. In particular you're not allowed to type in code and trying compiling and running it.*

Questions will mostly be similar in format to the ones in quizzes and the few non-opinion minute essays — some "what does this program do/print", some "write a program to do this task" — but somewhat longer and/or more difficult. There will also likely be a few multiple-choice or true/false questions.

## 2 Lecture topics to review

You are responsible for all material presented during lecture (up through user-defined data types), including the sample programs; the following is a list of major topics to review:

- Number systems — converting decimal to binary/octal/hexadecimal and vice versa.
- Data representation — basic idea of how negative integers and floating point numbers are represented and how this affects what can be represented (e.g., there is a largest `int`).
- Variables in C (types, declarations).
- Expressions and statements in C.
- Conditional execution in C.
- Functions in C — defining them, using them, using library functions.
- Recursive functions and loops in C.
- Arrays in C (mostly 1D, though there might be a question involving 2D arrays).
- "Order of magnitude" of functions.
- I/O in C — using library functions to read from and write to both stdin/stdout and files.
- Pointers in C.
- Strings in C.

- Command-line arguments.
- User-defined types, in particular ones using `struct`.

(Notice what's *not* on this list — everything past `structs`. I went through this material fairly quickly and don't think it's reasonable to ask you questions about it.)

## 3 Reading to review

You should have read, or at least skimmed, all of the assigned reading, but the focus will be on material presented or at least mentioned in class. (There is a lot of material in the textbook, the discussions of software engineering in particular, that I think is not crucial to the goals of this class.)