General Information

1. Prerequisites

It is expected that you have taken CSE 12/15L or another equivalent course in data structures. CSE 30 assumes that you have Java programming skills, and some exposure to Linux programming environment. We will use C language in this class though you are not required to have used C before.

2. Critical information, at a glance

You should read this entire syllabus. It is important. It may be the most important thing you read for this course. But here are the pieces of information you absolutely do not want to forget. I don't mean for this to sound scary, but so many students fail to read or understand these points, so I want to make them as clear as possible.

- Homework (PAs) is due by 11:59 pm on the due date. No late work will be accepted unless it is due to a documented emergency and the instructor of the course has to approve it before the due time.
- All reading assignments are due before the class for each session. We won't assign any points to readings.
- It is your responsibility to ensure that you have correctly submitted the correct code for your homework assignment. Incorrectly submitted assignments will be graded as is. We won't accept late work.
- All homework assignments must be done based on the instructions.
- All questions for the class should be posted to edsteam. Emails to the instructor should be about personal and confidential matters only.

3. What will I learn in this class?

- Write, Test, and Debug C and Assembly programs of medium complexity.
- Author C programs and functions with effective use of pointers, including at least the handling of pointers to pointers.
- Translate C functions, including recursive functions, functions using Strings, functions using user-defined types: structs, functions with parameters, and functions with pointers to Assembly and vice-versa.
- Analyze the memory layout of C and assembly programs including fully articulating the state of the stack and heap across function calls and after using malloc and free.
- Convert numbers between common computer science bases (binary, octal, decimal, and hex), convert to and from floating-point, and effectively employ bitwise operators in C and assembly.
- Effectively use debugging tools (gdb) and articulate the impact of basic compiler flags.

4. Textbook: freely available from UCSD library.

- **"Digital Design and Computer Architecture: ARM Edition - referred to as Harris in our reading"** by Sarah Harris and David Harris. This book should be accessible through UCSD's ebook library at **http://roger.ucsd.edu/record=b9288773~S9** **(http://www.google.com/url?q=http%3A%2F%2Froger.ucsd.edu%2Frecord%3Db9288773~S9&sa=D&sntz=1&usg=AFQjCNHUphlSdKg7kZEkjjjK1xu1HCAEDQ)**
- **Recommended book: "ARM Assembly Language : Fundamentals and Techniques, Second Edition - referred to as Hohl and Hinds in our reading"** by William Hohl and Christopher Hinds. This book can be accessed (at most three people a time) through UCSD's ebook library at

http://roger.ucsd.edu/record=b9498630~S9 (http://www.google.com/url?q=http%3A%2F%2Froger.ucsd.edu%2Frecord%3Db9498630~S9&sa=D&sntz=1&usg=AFQjCNGj7nF7poAmZQys78bv_J8rmGRs4A)