



**COLORADO SCHOOL OF MINES
ELECTRICAL ENGINEERING & COMPUTER SCIENCE DEPARTMENT**

**CSCI-410
Elements of Computing Systems
Spring 2014**

ECS-06A

Project ECS-06 differs from the reference ECS-06 (i.e., the one in the book) in two ways:

- 1) The project is broken into two pieces that are to be submitted separately. This is consistent with the recommended approach described in the book (Sec 6.3.3) in which you will first implement an assembler that does not deal with symbolic variable names or labels in the first stage (ECS-06A) and then add the ability to do so in the second stage (ECS-06B).
- 2) Your assembler must support the following extensions to the Hack assembly language presented in the book:
 - 2a) Your assembler must recognize 'X' comp mnemonics which are of the form 'XHH' in which HH is replaced with a two-digit, uppercase, zero-padded hexadecimal value. You convert this to a machine instruction by taking the low seven bits of this value and using them directly as the seven bits (a,c1...c6) in the C-type instruction. This means that the msb of HH is expected to be 0 and your assembler should throw an error (note – simply crashing is fine, but preference is given to dying gracefully including providing some kind of meaningful feedback).
 - 2b) Your code must produce an annotated copy of the original .asm file in a file with the same base name but with a .ann extension. The entire contents, including header and comments, should be copied across. You do not need to modify or add a header. After each instruction, you should add a comment starting at position 41 that contains the symbol-less version of the instruction followed by the binary sequence for the machine instruction as four space-separated groups of four. The machine instructions should be aligned. If there is already a comment on that line that extends past position 41, place the generated comment just after it (separate by a space), but remove any trailing whitespace from the original line before doing so.
 - 2c) Your code must append the Symbol Table to the end of the annotated source code file as a set of comments. You are free to choose your own format, but remember that the goal is to provide a debugging aid; hence, it should be very easy to determine if a symbol is a variable or a label and what address is associated with it. Note that this task is not required for ECS-06A and is here for completeness. It is required for ECS-06B.

File Submission

Be sure to adhere to the File Header Formats and the Submission Procedures.

If you wish, you may complete everything for both stages at once; but you must still submit separate and properly named ZIP files for each stage. Also, each ZIP file must extract to the proper directory, either 06A or 06B as appropriate. Other than that the contents of the two ZIP files can be identical.

In your file headers for ECS-06A may make them all ECS06 for ECS06A (or you can make them ECS06A, that is up to you). But in ECS06B you need to use ECS06B if there are ANY changes to the file compared to the one submitted for ECS06A.