

CSCI-410 Python Assignment - PY-03

The primary goal of this project is to familiarize you with using dictionaries and writing classes. At the end of this assignment, you should have a program that

- 1) Opens a .hack file and exports the content into an XML document.
- 2) Reads the XML document and produces an assembly source code file that will assemble and run.

You may continue working with your existing PY01/PY02 code, you may adopt an approach based, in part or in whole, on the Instructor's solutions, you may start entirely from scratch, or you may combine any and/or all of these. The choice is yours.

In the ECS coding projects, you will normally be given a specification for your program that details what each class is supposed to do and what the interface to it is expected to be. In this assignment, you are free to organize your work as you wish, but how well organized it is will be considered in grading.

The Instructor's solution to PY02 has been modified to provide a working example of classes and dictionaries. It consists of a file `cs410_inst_py.py` that contains two classes, one for managing a pair of input/output files and one that implements the `hack2xml` state machine. The dictionary is used to provide a reference between the string representing a state name and the function that processes information for that state. This is a slightly more advanced usage of a dictionary that is often seen, in which the dictionary is used to simply translate between a list of paired items. You will likely find it useful to have a few dictionaries, say one each to translate the `comp`, `dest`, and `jump` fields of the C-type instructions.

In addition to supporting the 28 `comp` mnemonics enumerated in Figure 4.3 of the text, your disassembler should support all possible 128 instructions. This will be done through the use of the `XHH` mnemonic in which `HH` is replaced by the two-digit zero-padded uppercase hexadecimal representation of `{a,c[1:6]}`, which is simply the seven bits of the C-Instruction starting with the fourth ('a') and ending with the tenth ('c6'). Note that all 28 of the official instructions have `XHH` equivalents. For instance, `M-D` is equivalent to `X07`, while `D+1` is equivalent to `X1F`.

I/O Filenames

As with PY02, the input file will be hardcoded as "machine.hack" and assembly output file will be "assembly.asm". The XML output file will be hardcoded as "machine.xml".

A few files will be made available for you to test your code against.

CSCI-410 Python Assignment - PY-03

Submission

In a manner similar to the ECS project, place all files needed for this assignment into a directory named PY03 and zip up the entire directory into a zip file of the form:

CS410_UserID_PY_03.zip

The main script – the one that that grader needs to run – must be named **py03.py**.

GRADING RUBRIC – 20 pts

- 5 pts Effort**
- 1 pt If the A-type instructions are not disassembled properly.**
- 1 pt For each standard mnemonic that is not disassembled properly.**
- 1 pt For each dest combination that is not disassembled properly.**
- 1 pt For each jump combination that is not disassembled properly.**
- 1 pt If the XHH mnemonic is not encoded properly.**

- 1 pt Incorrect header comments in Python script.**
- 1 pt Incorrect submission (filename, etc).**

The quality points will be deducted only up to a maximum of 15 pts. The format point deductions are above and beyond this.