CS1428 Lab 4h

# Name: Section:

In this lab you will begin to develop a VERY simple programming language for a VERY simple processor by using if-statements, switch-statements, and arrays. Today we will focus on arithmetic operations.

1. (15pts) The memory for our processor is small. It will only be 256 “cells” of storage with each cell storing an integer value. How is this *array* of storage declared?
2. (15pts) Your program will read programming instructions from a file. How do you get the name of, open, and read from an input file? (Hint: get the file name from the user)
3. (15pts) Each instruction in the file will take up one, and only one, line? The format of each line is:

**instruction-number instruction-data0 instruction-data1 instruction-data2**

The content of each instruction-data block depends on the instruction. For the arithmetic instructions, data0 is the index of the cell to write TO, data1 is the index of the LEFT operand, and data2 is the index of the RIGHT operand.

Write some code that reads in an instruction and then processes and performs that instruction if add is instruction-number 0, subtract is 1, multiply is 2, divide is 3, exponent is 4. (Hint: Use a switch statement, this is VERY much like the “calculator”)

1. (15pts) We also need to develop some **read** and **write** instructions to allow the program to input/output from the user, otherwise what good is it? Write the lines that you have to add to the above switch statement to implement **read** and **write** if the format of load instructions is indicated below: (Read takes the user input and puts it in the cell, write takes the value of the cell and prints it to the screen on a line by itself)

Read: **5 cell-index-to-store-the-value 0 0**

Write: **6 cell-index-to-display 0 0**

1. (15pts) Last, we need a way for the program to directly store constant values into its own memory. This is done by the **store** command. If store is implemented as instruction number 7 and the instruction data 0 is the location to store the data and instruction data 1 is the value to store (Instruction data 2 will always be zero), write the additional lines you have to add to the above switch statement to implement **store**.
2. (25pts) Using the segments you have written above, combine them into a single program (**lab6h.cpp**) that:
   1. Asks the user for the number of lines of “code” that will be entered
   2. Prompts the user line by line for the “code”
   3. Modifies the “memory” array according to instructions in the file
   4. You can test your programs against the following sample instruction files. (age.txt)

**(\*\*Make sure to include the standard header and to name the file correctly\*\*)**

**Turn in a hard copy of your program stapled to the back of this worksheet and a soft copy to homework upload.**