**CS 230 Project - Assembler**

*This project must be completed in groups of two.*

**Due Date:** See Canvas

**Purpose of Project**

You will gain experience with how instructions and data are represented in machine level.

**Description**

In this project you will create a very simple assembler program for Pep/9 Assembly language. This assembler will be able to assemble code from a text file and output the corresponding hexadecimal machine code in ASCII. For example, if your assembler reads a file containing the text below:

LDBA 0x0048, i

STBA 0xFC16, d

LDBA 0x0069, i

STBA 0xFC16, d

STOP

.END

it should output/print: D0 00 48 F1 FC 16 D0 00 69 F1 FC 16 00

Your team will just need to submit *one* copy (either member submits on Canvas). ***Please be sure to include a submission note stating the names of both team members.*** Be completely prepared to explain your code. Failure to explain will result in a zero grade.

Instructions:

1. The following instructions should be supported STBA, LDBA, STWA, LDWA, ANDA, ASLA, ASRA, STOP, CPBA, BRNE.
2. The following addressing modes should be supported: immediate, and direct.
3. Your program should be written in Java and should be named *pepasm*.
4. Your program should be run as follow:   
   *java pepasm file.pep*where file.pep is the name of the file containing the assembly code.

Tip: If you assemble a program with these commands using the Pep9 simulator, you will be able to find their corresponding codes.

Rubric

You are provided with files *program1.pep*, *program2.pep*, *program3.pep*, and *program4.pep*. You will be graded based whether your code is able to interpret and output the right machine code for each one of these, as follows:

* Correct output for program1.pep: 50 points
* Correct output for program2.pep: 30 points
* Correct output for program3.pep: 10 points
* Correct output for program4.pep: 10 points