**CSE/EEE230 Assignment 1**

**Due September 6, 5PM.**

This assignment is designed to introduce you to the MIPS assembly and the MARS simulator..

1. Open the text editor and type in the following. Replace the string “Your name” with your actual name. Then save your work as assign1.asm

# CSE/EEE230 Fall 2016

.data

val1: .word 1

val2: .word 2

val3: .word 3

name: .asciiz “Your name\n”

.globl main

.text

main:

1. Add the MIPS assembly language instructions after the main for the following actions. Do not skip or change the order of the steps. Note that the variables val1, val2, and val3 represent addresses or locations in memory in the data segment. You may use other registers as needed.
2. Set a base register ($t0) to the start of the data segment 0x10010000
3. Print your name on its own line using the correct syscall command. You will need to determine the displacement from the beginning of the data segment to your name.
4. Initialize the following registers
	1. Put the value 7 into register $s0
	2. Put the value 8 into register $s1
	3. Put the value -3 into register $s2
5. Store the values into memory
	1. Store the value in $s1 into the memory labeled “val1”
	2. Store the value in $s2 into the memory labeled “val2”
6. Calculate and store into memory. Do not change the values of the registers in this process.
	1. Calculate the value of $s0 - $s1 + ($s2 – 5)
	2. Store this result in the memory labeled “val3”.
7. Update the registers to the following values. Do not change the values in memory
	1. Change the value in $s1 to 10
	2. Change the value in $s2 to 5
8. Exchange or swap the values in val1 and val2. This will require several steps.
9. Set the value in $s0 to –$s0
10. Calculate the value of $s0 + $s1 + $s2 and print the result. Do not change the values of $s0, $s1 and $s2
11. Stop the program by using a syscall with the command 10.

Specifics:

* Use only the instructions covered to date plus the syscall command.
* You will need to calculate the addresses for val1, val2 and val3. Remember that the .data segment starts at 0x10010000.
* **Do not use pseudo-instructions**. This includes la and other pseudo instructions. Use only the core instructions from the reference sheet. To prevent this, make sure that the “Permit extended (pseudo) instructions and format option” under Settings is **cleared**.

Documentation:

* Comment the beginning of your programs with your name, class, and assignment number.
* Comment each step (above).
* Comment every instruction.

Hints on getting started:

* Create an .asm file using the MARS simulator with the content from part a (above).
* Add comments at the beginning as required.
* Solve step 1. Assemble and check your work.
* If step 1 is correct, then solve step 2. Continue in this incremental fashion until all steps are completed.

Assignment submittal:

* Check your work using the simulator to make sure that you have the correct results at the end in memory, the registers and output.
* Upload your assembly language program (.asm file) using this link on Blackboard.

Notes on grading

* You must show all steps in the program as required in the instructions. Do not just print the result.
* Your results will be checked for the following
	+ correct output (name and result)
	+ correct ending values in memory
	+ correct ending values in registers
	+ no use of pseudo-instructions