# Project - Chapter 3: Simple Arithmetic

## Possible Points: 30

## The Problem

Let's do some arithmetic with different data types!

A. Create a Linux-based assembler language program (nasm) which:
    1. Defines these variables:
        1. A: A single byte
        2. B: A word
        3. C: A double word
        4. D: A double word
    2. Using the eax register (and its sub-registers), process the following equations (ONLY using the **mov**, **add** and **sub** assembly keywords):
        1. A + (B + C) = D
        2. (A + C) - B = D
    3. Using the linux function library, print a string describing each equation, the values in each variable, and then the answer
B. Create a Windows-based assembler language program (masm) which does everything that your linux program does (A), except, use the appropriate MASM keywords and the Irvine print functions

## Assumptions

Here are some of the things I assume you know how to do attending class and reading the text(s).

- You understand how to write, assemble and link both a Linux assembly language program and a Windows assembly language program.
- You have attended class and read the text and understand all of the data types which you can define within an assembly language program
- You understand how to create byte, words and double-word variables in assembly language and use the appropriate CPU registers
- You understand how to move data into different CPU registers as well as move them out into array positions in memory

## Data to be Used

- Just what you define in your program.

## Sample Output and the Solution

- None

## Hints/Tips

- None

## Extra Credit

- None

## How to Submit

- Assignment submissions will only be accept via Canvas.
- Please compress/zip the following files/folders:
    - The project folder for your Linux based assembly language program using NASM compressed as a zip file. This should include everything including the build scripts
    - The project folder for your Windows based assembly language program created using Visual Studio 2015 or 2017 compressed as a .zip file.

## Grading Criteria

| | |
|---|---|
| Program Does not Assemble or Link | -30 |
| Each missing item as defined in “The Problem” section of this document | -5 |
| Late | -30 |