CSUS

COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

Department of Computer Science

CSC 35

Spring 2021

Dr. Ghansah

**Lab 1: Introduction to Microsoft Assembler (MASM), Microsoft Visual Studio (MVS), and Debugger**

**Purpose:** This Lab is intended to help you get familiar with the MVS/MASM programming environment in our remote lab. The instructions on 1) how to access the remote labs and 2) how to set up MASM project on Visual Studio 2019 in the remote lab are both in Canvas. However, you are free to do this and subsequent assignments on your own local machine if you have Visual Studio configured to use MASM installed on it.

For instructions on how to install Visual Studio and configure it to run a MASM project please see the Irvine website whose URL is in the course syllabus.

This lab is intended to introduce you to assembly language. Assembly language is called a low-level language because it is close to machine language in structure and function. Each assembly language instruction corresponds to one machine instruction (a one-to-one correspondence) although the speed (number of clock cycles needed) for each instruction varies. In contrast, single statements in high-level languages such as C, C++, Java, and Python are translated into multiple machine instructions. (e.g. an example of an assembly language instruction would be "MOV AL, 0FFh", which will load the AL register with the 8 bit hexadecimal value FF. Similarly, "MOV AX, 00FFh", which will load the AX register with the 16 bit hexadecimal value 00FF. Finally, "MOV EAX, 000000FFh", which will load the EAX register with the 32 bit hexadecimal value 000000FF.)

You're going to be working with MASM within programming environment from Microsoft called Visual Studio. In this integrated environment you can write assembly code, assemble the program, run the program and enter into the code debugger.

For purposes of labs in this course, Visual Studio is the recommended place to code, run and debug your programs.

During today’s lab, you will learn how to assemble/build your program, find compiler errors, and learn how to "step" through the program line by line by using the Visual studio debugger.

***Procedure:***

Using the instructions in the document with the file name “masm setup and debug instructions” (this is on Canvas. Note that if you are running MASM and Visual studio on your own machine some of the steps in the file should be skipped).

a) Assemble, compile, and debug the program provided in a .asm file on Canvas.

b) Learn how to set break points, step through the program, and watch changes in register values all within the debugger.

**Testing:** Be sure to test your program and make sure it works before you submit it to your lab instructor on CANVAS as specified below.

**Demonstration**: Demonstrate your program by providing screen shots showing the steps, especially assembly and debugging steps including changes in register values. Normally, the lab instructor will assemble and run the documented source code you upload to CANVAS. However, since the program was given to you the instructor will check whether your report contain the proper steps.

**Submission:** Submit electronic copy of your program to CANVAS including a well *documented program (source code)* and output (screen shots). **Filenames must be according to the format specified in the syllabus**