**CSCI 360-1 Assignment 1 - Your First Mainframe Program Spring 2019**

**10 points**

Before you begin this assignment, you must first allocate the PDSE (Partitioned Data Set Extended) data set described in Section 12.1 (Chapter 12, Section 1) in the CSCI 360 Course Notes.

For 10 points, create and edit a new member of your Assignments PDSE and name the new member ASSIGN1. You can start editing referring to Section 12.2 in the CSCI 360 Course Notes.

Type the JCL (Job Control Language) and Assembly language program below that begins with the line

//KC03nnnA JOB ,'your name here',MSGCLASS=H

and ends with the line with only two forward slashes ( // ) into your PDSE member named ASSIGN1.

When you have finished and made the editing change to it described in red in the "doc box" below, you will have what's known as a "job" that can be submitted for execution on the Marist University mainframe. By the way, please do not type the text in red or the line that says "Note that this documentation box does not go past column 65!" into your finished program.

//KC03FB5A JOB ,'H ATTARWALA',MSGCLASS=H

//JSTEP01 EXEC PGM=ASSIST

//STEPLIB DD DSN=KC00NIU.ASSIST.LOADLIB,DISP=SHR

//SYSPRINT DD SYSOUT=\*

//SYSIN DD \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* ASSIGNMENT 1 - YOUR FIRST MAINFRAME PROGRAM \*

\* \*

\* DATE DUE: 01/18/2019 \*

\* \*

\* **Replace KC03nnn above with your KC-ID assigned to you by** \*

\* **your instructor. DO NOT LEAVE OUT THE CAPITAL LETTER A AT** \*

\* **THE END OF YOUR KC-ID! Also, put your name in all capital** \*

\* **letters in the first line where it says 'your name here'.** \*

\* \*

\* Note that this documentation box does not go past column 65! \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MAIN CSECT

USING MAIN,15 ESTABLISH ADDRESSABILITY ON REGISTER 15

LA 3,12 LOAD 12 INTO REGISTER 3

LA 9,5 LOAD 5 INTO REGISTER 9

AR 3,9 ADD REGISTER 9 TO REGISTER 3

XDUMP , DUMP THE REGISTER CONTENTS

BR 14 RETURN TO CALLER

\*

LTORG LITERAL ORGANIZATION

\*

END MAIN

/\*

//

It is extremely important that you not save any blank lines before or after what is presented above. The columns on each line and where things appear here are also extremely important.

In the Assembly language, columns 1, 10 and 16 are significanct. Labels like MAIN begin in column 1, instruction mnemonics like LA (Load Address) begin in column 10 and instruction operands like 3,12 begin in column 16. All letters must be capital letters too!

Type the word SAVE on the command line and press ENTER frequently so that you will not lose any of your work if you accidentally sign yourself off or your computer or connection crashes.

Assembly language, henceforth known as "Assembler," is a very unforgiving language that requires close attention to detail. The lines of the job above that begin with at least one forward slash (/) are lines of JCL and are ***not*** Assembler code.

We will use the same JCL for all of our programming assignments and in our examples for the remainder of the semester. There will be minor changes with lines of information added to the end of the job but that will come later.

The JCL above surrounds the Assembler program and is used on the mainframe to tell the operating system what type of work you are trying to perform. In this case, we are compiling an Assembly language program and, if that is successful, we are running it. We do not call it compiling, though, when using the Assembler; it is called assembling rather than compiling.

In summary, the first line of the JCL tells the mainframe operating system, z/OS, that it is **your** job with your KC-ID. The word JOB tells z/OS that the first line is the beginning of a new "job." Note the required space between JOB and the comma. Inside the single quotes (apostrophes which we will call "ticks" or "tick marks"), you can put up to 20 characters of your choice. For this class, please put your first and last name or, if too long, your first initial, a period, and your last name. The MSGCLASS=H tells the Marist system where to place your job's output when it finishes. H is the "held" class and your job's output will be placed in the held output queue.

The line with EXEC PGM=ASSIST tells the Marist system that you want to execute the program object named ASSIST. A program object is essentially the "executable" on the mainframe. (You will also hear executables called "load modules".) ASSIST is a learning version of the mainframe Assembler that assembles, or "compiles," your program and actually executes it too but only if it assembles with no errors.

When you are done typing and are ready to run your job and view the results, first save your work and then type SUB on the command line and press ENTER. You can then refer to Section 12.3 of the CSCI 360 Course Notes.

When you are satisfied that the results are correct, you must then use mar\_ftp.exe to download a .txt file version of your job output from Marist. Refer to Section 12.4 in the CSCI 360 Course Notes.\*

Now, once you have the .txt file downloaded to your own laptop or PC, open the .txt file and review it from top to bottom and compare it to the sample output provided. Yours should look pretty much the same except for the highlighting. We will discuss this output in class and the highlighting will be explained.

When you are satisfied, submit a copy of the .txt file on Blackboard by the time and date it is due.

\*If you find that mar\_ftp.exe is not connecting or just not working for you, go to Section 12.5 of the CSCI 360 Course Notes and use FileZilla to download your output.