

To Start

Before you begin this assignment, you must first allocate what is known as a PDSE (Partitioned Data Set Extended). It is a mainframe data set that will hold your assignment files for the semester. We will cover how to do this during the first week of classes. You will be guided through allocating your partitioned data set extended (PDSE) from within **IBM Developer for z/OS (IDz)**. You will also be shown how to create a new "member" of the PDSE into which you will copy and edit the program below. Please note that you will not create a new PDSE for each assignment, just a new *member* of that PDSE for each successive assignment this semester.

Note that there is also a video titled **How to Use IDz and Download Job Output.mp4** under the first item under Blackboard menu item *Software* that will lead you through getting signed on to the Marist mainframe using IDz and allocating your assignment PDSE which we will call your ASSIGNS PDSE. The data set's full name will be:

KC03nnn.CSCI360.ASSIGNS where nnn represents the last 3 characters of your Marist ID

You will actually need to create two PDSEs. One named ASSIGNS and another named SUBMTTD. You can use the video mentioned above to allocate the ASSIGNS PDSE first and then watch it again to allocate the SUBMTTD PDSE substituting SUBMTTD for ASSIGNS the second time through the allocation. This data set's full name will be:

KC03nnn.CSCI360.SUBMTTD also where nnn represents the last 3 characters of your Marist ID

For the remainder of this assignment, you will only use the ASSIGNS PDSE. The use of the SUBMTTD PDSE is described in the *Course Syllabus* and will also be discussed in class.

The Assignment

For 25 points, create and edit the new member of your ASSIGNS PDSE. You should have named the new member ASSIGN1. When you are ready to begin editing that member, type or copy and paste 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 is known as a "job" that can be submitted for execution remotely on the Marist University mainframe. By the way, you can delete the lines with the red text from your PDSE member if you choose to do so.

It is important that you begin in column 1 for lines 1 through the line with MAIN on it and be sure that the final two lines begin in column 1. In Assembler, columns make a difference so do not change the position of anything in this code.

Once you have the JCL and program typed or copy and pasted into the PDSE member named ASSIGN1, make changes to it as described in the doc box in red below.

```
//KC03nnnA JOB ,'your name here',MSGCLASS=H
//JSTEP01 EXEC PGM=ASSIST
//STEPLIB DD DSN=KC00NIU.ASSIST.LOADLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
*****
*
* CSCI 360-n ASSIGNMENT 1 current semester *
*
* DEVELOPER NAME: your name goes here *
* DUE DATE: the assignment due date goes here *
*
* Replace nnn in KC03nnn in the first line 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! Then, put *
* your first initial, a period and your last name in all *
* capital letters in the first line where it says 'your name *
* here'. *
*
* Change CSCI 360-n to reflect your section number. Change *
* current semester to reflect the current semester like *
* SUMMER 2023. Fill in your name as the developer and enter *
* the assignment's due date. *
*
* Note that this documentation box does not go past column 65! *
*
*****
*
MAIN CSECT
      USING MAIN,15 ESTABLISH ADDRESSABILITY ON REG 15
      LA 4,13 LOAD 13 INTO REG 4
      LA 8,6 LOAD 6 INTO REG 8
      AR 4,8 ADD REG 8'S CONTENTS TO REG 4'S
      XDUMP , DUMP CONTENTS OF ALL 16 REGS
      BCR B'1111',14 UNCONDITIONAL RETURN TO CALLER (OS)
*
      LTORG LITERAL ORGANIZATION
*
      END MAIN
/*
//
```

It is very 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 very important.

In the Assembler language, columns 1, 10 and 16 are significant. 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!

Assembler language, or "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, or Job Control Language, and are *not* Assembler code.

We will use the same JCL for all our programming assignments and in any 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, z/OS, what we are trying to accomplish. In this case, we are compiling an Assembly language program and, if that is successful, we want to execute, or "run", 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 the mainframe operating system, 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 mainframers 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, a space 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, or executable, named ASSIST. A program object is what we call an "executable" on the mainframe. (You will also hear executables called "load modules" but it is old terminology.) 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 you are ready to run your job and view the results, first save your work and then type SUB on the command line – the long rectangle at the bottom of the editing window – and press ENTER. After you have closed out of the member and then come back into it later, you will no longer see that command line but can instead either right click the mouse in the editing area of IDz and then click **Submit** near the bottom of the choices presented or, more directly, press **Ctrl+Alt+M** and it will submit the program you have open.

After you submit the job, you will see the job number assigned by JES (the Job Entry Subsystem) displayed. You might see the **Getting Started** tab just below the editing window. If it is there, you can simply close it. To be efficient, though, you need to see the **Remote System Details** tab. If it is not there, go to the toolbar and click on **Window → Show View → Remote System Details** and you should start seeing it from then on.

Once the tab opens, you should see a list of your job or jobs. Note that the output in the tab needs to be refreshed periodically. To do so, right mouse click in that window and click **Refresh**. Another way to view your jobs is to click on **JES** in the file structure window on the right and click **My Jobs**. Select the job you want to review and double click on it.

When you are satisfied that the results are correct, you must get a copy of your output down from Marist onto your own machine and convert it into a text document named ASSIGN1.txt (change the name to ASSIGN1.txt, if necessary) to submit it on Blackboard. You will be shown how to do this within the first few days of class using the Retrieve Jobs website which not only downloads your Marist output but also converts it to a .txt file. Make sure the downloaded file is named ASSIGN1.txt for this assignment, to ASSIGN2.txt for the next assignment, etc.

Once you have the text document on your own machine, it is VERY important that you do NOT change anything **in** the text document you plan to submit. **If you do, it is considered cheating!**

Submit a copy of your ASSIGN1.txt file on Blackboard by the time and date it is due. Also, copy your ASSIGN1 member from your ASSIGNS PDSE into your SUBMTTD PDSE and be sure you do not open and/or change it! The date and time stamp of the member of the SUBMTTD PDSE are important.