CSCI 465/680-J9 **Assignment 1 – Copy and Execute JCL** **Fall 2019**
10 points

Introduction

The purpose of this short extra credit assignment is to 1) give you experience (or more experience) using the TSO/ISPF Utilities option to create a PDSE, or partitioned data set extended, for you to store your semester's assignment files on the Marist mainframe's direct access storage devices (DASD), 2) give you experience editing, saving and submitting a member of the same PDSE using TSO/ISPF Edit option, and 3) give you experience reviewing the output of your submitted "job." Note that much of this will be shown to you in class several times during the first week and beyond.

Reference Material

The following are the first four sections found in *Chapter 12 - Working on the Marist Mainframe* in Course Notes on Blackboard and will help you complete this assignment:

12.1 Allocating Your Assignment PDSE
12.2 Editing in ISPF
12.3 Viewing Results in SDSF
12.4 Using mar_ftp.exe to Download Marist Job Output

Instructions

First, after signing on to Marist as shown to you in class, go to Utilities option - Option 3.2 from TSO's ISPF Primary Option Menu (ISPF = Interactive System Productivity Facility). Here you will create a PDSE, or partitioned data set extended, that will store your assignment files for the entire semester. In a PDSE, the separate files are known as "members." Please name this data set exactly as follows:

KC0nnnn.CSCI465.ASSIGNS *Graduate students *also* use CSCI465, please.

In the name, KC0nnnn is your Marist ID as assigned to you in the Excel spreadsheet named KCTR Marist Student IDs under Marist Student KC-IDs in the Course Documents on Blackboard. (Naming your data sets as instructed here enables the TA and instructor to review your progress and, when necessary, to quickly and efficiently copy PDSE members in order to assist you in debugging and/or correcting mistakes in assignments.

Allocate your ASSIGNS PDSE with space units of TRKS (tracks), a primary quantity of 10, a secondary quantity of 10, 10 directory blocks, record format of FB (fixed blocked), a record length of 80 (bytes), and a block size of 3200, and, near the bottom of the screen, a data set name type of LIBRARY. This PDSE is for your assignments this semester. Press enter and you should see a message in the upper right hand corner stating Data set allocated. If you suddenly see a screen of red type stating something about the VTOC being full, ignore it and just press Enter again.

Note that YOU are responsible for backing up this very important data set on a regular basis, i.e., frequently.) This can be done using Option 3 – Utilities and will be demonstrated for you in class.

Second, go to Option 2 – Edit to open a new member of the PDSE you just created. You will name the new member ASSIGN1. Type in the JCL provided in blue below, make the changes to it as described, and then submit the job as shown to you in class. Review the output in SDSF and, if the job was successful, you

should have a second PDSE created that will hold programs, i.e., source code. Please note that EACH assignment you write this semester, name the PDSE member for that assignment appropriately, such as ASSIGN1, ASSIGN2, ASSIGN3, ASSIGN5A, etc.

You do NOT need to type all of the documentation shown below into your member but you must include the first five lines beginning with CSCI 465/565 and ending with DUE TIME:. Be sure to change *your last name* in the first line of the JCL to…guess…your last name and, in the documentation box, change *your name* to your own name and change *nnnn* to the last four digits of your Marist KC-ID.

Type the Following in the ISPF Editor

```
//KC0nnnnA JOB ,'your last name',MSGCLASS=H
//*
//*************************************************************
//*                                                           *
//*  CSCI 465/565 - FALL 2019                                 *
//*                                                           *
//*  ASSIGNMENT 1 - COPY AND EXECUTE JCL                      *
//*                                                           *
//*  PROGRAMMER:  your name                                   *
//*                                                           *
//*  DUE DATE:  08/30/2019                                    *
//*  DUE TIME:  11:59 PM                                      *
//*                                                           *
//*  The purpose of this job is to allocate a new PDSE, or    *
//*  partitioned data set extended, that holds load modules.  *
//*  There are two job steps, both running IEFBR14.  The      *
//*  first will delete any possible previous version of the   *
//*  PDSE named KC0nnnn.CSCI465.LOADLIB.  The second creates  *
//*  a new PDSE of the same name.  IEFBR14 is one of the      *
//*  most used of all the IBM utilities available.            *
//*  IEFBR14's source module consists only of a CSECT with    *
//*  a single BR 14 instruction and really accomplishes       *
//*  really nothing as a stand-alone program BUT, as we       *
//*  will soon find out, it has MANY uses for allocating      *
//*  (creating), cataloging, deleting, and un-cataloging      *
//*  data sets.  In this two-step job, you are simply         *
//*  executing IEFBR14 twice.  The first step deletes any     *
//*  previously-created PDSE that might exist from a prior    *
//*  successful run of the job and the second allocates, or   *
//*  creates a new one.  This PDSE is the second of the       *
//*  three PDSEs that we will need this semester.             *
//*************************************************************
//*
//JSTEP01  EXEC PGM=IEFBR14
//*
//*  The following DD card deletes an existing PDSE if it
//*  already exists.  If it does not exist, the data set is
//*  first allocated and then immediately deleted.
//*
//DD1      DD DSN=KC0nnnn.CSCI465.LOADLIB,
//            SPACE=(TRK,(1,1,1)),
//            DISP=(MOD,DELETE,DELETE)
//*
//JSTEP02  EXEC PGM=IEFBR14
```

```
//*
//*  The following DD card allocates a PDSE to hold your
//*  program load module files.
//*
//DD1      DD DSN=KC0nnnn.CSCI465.LOADLIB,
//            DCB=(RECFM=U,LRECL=0,BLKSIZE=4096),
//            DSNTYPE=LIBRARY,
//            SPACE=(CYL,(9,5,5)),
//            DISP=(NEW,KEEP,DELETE)
//
```

Note: In each assignment – beginning with Assignment 2 – it is very important to follow the JCL documentation standards as described in *Chapter 1 - Coding and Documentation Standards* found in Blackboard's Course Notes. Always use the documentation box ("doc box") shown above and, of course, change it as necessary to reflect the current assignment.

Once again, it is YOUR responsibility to be sure that you regularly copy (back up) all of your PDSEs throughout the semester. The loss of a PDSE due to programmer error is NOT an excuse for being granted an extension on an assignment or project!

Note that, upon completing this assignment successfully, you will have allocated two PDSE data sets. One will hold your assignments and the other will hold the executable load modules of your programs. (We will talk a LOT more about what these terms mean later.) You will also have allocated one of the two PDSEs using TSO/ISPF and the second using batch JCL to run a utility named IEFBR14 that allocated it. TWO DIFFERENT METHODS!

When you are ready to sign off of Marist, do so EXACTLY the way shown in class. If you simply close the Vista TN3270 window, you will cause problems with your account and even possible with your assignment data set. Sign off correctly and exactly like this EVERY TIME.

How to Submit Your Assignment

Use mar_ftp.exe to get .txt file version of the output from the Marist mainframe's output queue down onto your own laptop or PC. **Be sure to open the .txt file and inspect it from top to bottom EACH time you download using mar_ftp.exe. Be sure that all of your output is there!** Then, when you are sure it is, submit the .txt file on Blackboard as directed.

Once again, be sure to open the .txt file and inspect it from top to bottom EACH time you download using mar_ftp.exe. Be sure that all of your output is there! This is critical because, if you turn in a file with any missing output, you will earn a 0.