CSCI 465/680-J9 **Quiz 1** **Name____KEY_____**
Fall 2019 **25 points** **Wednesday, 09/25/2019**

Please print very carefully, lines things up clearly and use upper- and lower-case letters wherever necessary. When adding parameters to DD cards, please write one per line.

1) Write a complete JCL job step named `JSTEP03` that executes a load module named `CALCSHRS`. The program requires a maximum of `4096 KB` of main memory and should be allowed to execute only if one or more previous steps ends with a return code of `4` or less. `CALCSHRS` is an existing load module found in `NEWACCTS.SYSTEM.LOADLIB`, it reads records from flat file `NEWACCTS.DAILY.TRANS.G008V00` with ddname `TRANS` and writes records to standard output with ddname `REPORT`. No `JOB` or `SYSUDUMP` cards, please! (8 points)

```
//JSTEP03  EXEC PGM=CALCSHRS,REGION=4096K,COND=(4,LT)
//*
//STEPLIB  DD DSN=NEWACCTS.SYSTEM.LOADLIB,DISP=SHR
//*
//TRANS    DD DSN=NEWACCTS.DAILY.TRANS.G008V00,DISP=SHR
//*
//REPORT   DD SYSOUT=*
```

2) Write a complete JCL job step named `JSTEP01` using the appropriate IBM utility used in your Assignment 1 and discussed in class to allocate a PDSE named `NEWACCTS.SYSTEM.SOURCE`. Allocate in cylinders with 10 primary and 5 secondary. It will need 6 directory blocks and fixed length 80 byte records with a blocking factor of 20. NOTE THAT THE DATA SET MAY ALREADY EXIST AND, IF SO, YOU WOULD NOT WANT TO DELETE IT! (8 points)

```
//JSTEP01  EXEC PGM=IEFBR14
//*
//DD1      DD DSN=NEWACCTS.SYSTEM.SOURCE,
//            DCB=(LRECL=80,BLKSIZE=1600,RECFM=FB)
//            SPACE=(CYL,(10,5,6)),
//            DSNTYPE=LIBRARY,
//            DISP=(MOD,KEEP,KEEP)
```

3) Write a complete instream Assembly step. For the Assembler program itself, simply write: *Assembler code goes here*. List all of the ddnames and the `DD` itself. For the remainder of the `DD` card, provide a short but accurate description of what the `DD` represents for the Assembler. You may assume the Assembler is in the Linklist. Note: If you can, write the whole `DD` card for extra credit! To get credit for it, though, it must be exactly correct according to the *CSCI 465/680-J9 Course Notes*. (9 points)

```
//JSTEP01  EXEC PGM=ASMA90,PARM=ASA
//*
//SYSLIB   DD DSN=SYS1.MACLIB,DISP=SHR   or   Assembler macros
//*
//SYSIN    DD *
      Assembler code goes here
/*
//SYSLIN   DD DSN=&&OBJMOD, etc.   or   Output object module
//*
//SYSPRINT DD SYSOUT=*   or   Assembler listing and messages
//*
//SYSUT1   DD SPACE=(CYL,(1,1))   or   Scratch pad for Assembler
```