1. The three types of JCL statements.

**JOB, EXEC, DD**

2. How to use the three types of JCL statements.

**JOB – Information that schedules the job, which processor will be used, what the job’s priority is, and who the user is.**

**EXEC – beginning of a new step in the program. (module)**

**DD – identifies any data sets that are to be used in that step.**

3. How to refer in JCL to an already-allocated, cataloged data set.

**IEFBR14 – or new data set Program**

4. How to use IEFBR14 to allocate or delete data sets.

5. How to allocate a new PDSE in JCL.

Example:

**//JSTEP01 EXEC PGM=IEFBR14**

**//\***

**//DD1 DD DSN=KC01234.CSCI465.LOADLIB,**

**DCB=(LRECL=125, RECFM=FB,BLKSIZE=1250),**

**SPACE=(TRK,(5,10,5)),**

**DSNTYPE=LIBRARY,**

**DISP=(NEW,KEEP,DELETE)**

6. How to allocate and use a temporary data set in JCL.

**Use two (ampersands) && for the name, then up to 1-8 characters, only one node in the temporary data set name.**

**Must specify amount of space on the storage unit.**

Example: **RECSOUT1 DD DSN=&&TEMP1,**

**SPACE=(TRK,5,1)),**

**DISP=(NEW,PASS,DELETE)**

**~~~ in the next job statement: ~~~**

**RECSIN2 DD DSN=&&TEMP1,DISP=(OLD,DELETE,DELETE)**

7. How to distinguish between the execution of a program object (load module) vs. the reference to a procedure.

**//stepname EXEC proc name (or PROC=proc-name)**

Referenced cataloged procedure to execute

**//stepname EXEC PGM=prog-name**

Program’s load module

8. How to use the COND= parameter.

**Used to conditionally execute a job step. The return code is used in comparison to the previous step.**

**0 = ran successfully, 4 = completed with warning, 8 = error, 12 = serious error, 16 = severe error**

**If the condition returns FALSE, execute the step.**

9. How to use the TIME= parameter. **TIME=(m,s)**

**Specifies the amount of time the entire job will need.**

**m - number of minutes (integer)**

**s - number of seconds (integer)**

10. How to use the SPACE= parameter in detail.

**Specifies the amount of space on the storage unit required for the data set being allocated or created**

**SPACE=(unit,allocation)**

**units : TRK (tracks), CYL (cylinders), n (number of bytes to be allocated)**

**Allocation: the amount of storage in the format (p,s)**

P - **Primary - takes \_\_ number of tracks/cylinders/n bytes off the bat**

S - **Secondary, allocates \_\_ bytes up to 15 times**

Example: **SPACE=(TRK,(5,5))** - FLAT-FILE

11. How to use the DISP= parameter. Ex: DISP=(NEW,KEEP,DELETE)

**The disposition of the data set being referred to by the DD card.**

**DISP=(status,normal,abnormal)**

Status - current status of a data set at the **beginning of the job step**.

**NEW - Data set is to be created (default)**

**OLD - The data set exists, locked for access by the program running in the current job step**

**SHR - Data set has already been created and other programs may access it**

**MOD - The data set can have records added to it. Created if the data set does not exist.**

Normal/Abnormal:

**KEEP - Keep data set at the end of the job step.**

**PASS - Data set is to be passed to a later step within the job \*MUST BE USED FOR TEMPORARY DATA SETS\***

**DELETE - The data set is to be deleted at the end of the job step**

**CATLG - The data set is kept and cataloged at the end of the job step.**

**UNCATLG - The data set is kept but uncataloged at the end of the job step.**

12. Know how to write an Assembler step.

Module name: **ASMA90**

Required parms: **PARM=ASA**

-do not need STEPLIB, but we do require SYSLIB

**//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**//JSTEP01 EXEC PGM=ASMA90,PARM=ASA**

**//SYSLIB DD DSN**=SYS1.MACLIB,DISP=SHR

**//SYSIN DD \***

\*Assembler code goes here via instream data\*

**/\* (end of instream data)**

13. Know how to write a COBOL Compiler step.

14. Know how to write a Binder step.

15. Any other parameters not mentioned here that were discussed or used in class or on assignments.

16. The names of the three common modules we use.

COBOL – **IGYCRCTL,** ASSEMBLER – **ASMA90,** Binder - **HEWL**

17. The DD cards required by each of those three common modules and what each represents.

18. The three different ways to indicate in-stream data.

**DD \*, DD DATA, DD DATA,DLM=’xx’**

19. Know how to compute a data set’s blocking factor.

20. Know how to write a complete job with a JOB card and any IEFBR14, Assembler, COBOL Compiler, Binder and/or fetch steps as directed.