z/OS Introduction and Workshop Lab Systems

LAB 1
=====

Using TSO, SDSF, SSH and FTP
====================================

Replace all references to ###### with your assigned userid.

TSO
====================================

From ISPF Primary Option Menu command line
 =x <enter>

If screen appears requesting how to handle log
data set, then type '2' <enter> to delete

Observe TSO Ready prompt

Note: *** requires <enter> to continue

Type and <enter> the following commands from TSO Ready
 time
 profile
 listc
 help
 help profile
 help listc
 netstat
 profile nomsgid
 netstat
 profile msgid
 netstat
 profile nomsgid
 netstat
 netstat home
 netstat socket
 netstat co
 netstat tel
 netstat byte
 netstat dev
 netstat gate
 netstat route
 netstat help
 send 'i am testing the tso send command' u(######)
 help send
 ispf
 =sd

SDSF
==========

From SDSF Primary Option Menu type

prefix * <enter>
owner * <enter>
dest <enter>
st <enter>

Observe the status of all JES entries are displayed

prefix #######*;st <enter>

Observe the status of all ###* JES entries are displayed

prefix *;owner ######*;st <enter>

Observe only ###### JES entries are displayed

log <enter>

Observe the system log (syslog) entries

f ###### prev <enter>

Observe syslog entry containing your assigned ###### .

ulog <enter>
/D IPLINFO <enter>
/D PARMLIB <enter>
/D M=CPU <enter>
/D M=STOR <enter>
ulog <enter>

Observe content of SDSF user log (ulog)

F3 (function key 3) to exit SDSF
LOGOFF <enter> to terminate TSO session

SSH to z/OS Unix System Services
==========

Use either PuTTY, Teraterm, workstation command prompt

ssh *lab_ip_address* <enter>

Enter assigned ###### and password

From Unix System Services command prompt
enter the following:

```
id
uname -Ia
onetstat
echo test record > test
cat test
ls -al
env
cat /etc/profile | more
 ....space bar will continue with print display

exit
```

FTP
==========

From workstation command prompt
ftp *lab_ip_address* <enter>

Enter assigned ###### and password

From ftp command prompt enter the following:

```
pwd
dir
ls
```

Observe difference between dir and ls

```
cd /u
pwd
dir
ls
```

Observe difference between dir and ls which lists
unix files in /u subdirectory

```
quote site filetype=jes
quote site jesstatus=all jesjobname=* jesowner=*
dir
```

All entries in JES spool are displayed

```
quote site jesowner=######
  ....where ###### is your assigned userid
dir
```

Observe difference. Only JES entries owned by your userid will be listed.

```
quote site filetype=seq
pwd
ls
```

Observe that this has changed mode back to files

```
quit
```

LAB 2
=====

ISPF, ISPF Edit, JCL and SDSF
============================

1. Edit and submit JCL member to JES for execution
   a) TSO/ISPF =3.4
   b) Enter ###### in the Dsname level field
   c) Enter 'e' to the left of ###### .JCL
   d) Enter 's' to the left of JCLTEST member
   e) Enter submit
   f) Observe and note JES JOB number.

2. Review JCL JOB output in SDSF
   a) Enter =sd;owner ###### ;st
   b) Enter '?' to the left of JES job number noted above
   c) Enter 's' to the left of each DDNAME in the output spool
   d) Observe content of JESMSGLG ..job messages
   e) Observe content of JESJCL ..job JCL
   f) Observe content of JESYSMSG ..job system messages

3. Edit and submit JCL member to JES for execution
   a) TSO/ISPF =3.4
   b) Enter ###### in the Dsname level field
   c) Enter 'e' to the left of ###### .JCL
   d) Enter 's' to the left of JCLERROR member
   e) Enter submit
   f) Observe and note JES JOB number.

4. Review JCL JOB output in SDSF
   a) Enter =sd;owner ###### ;st
   b) Enter 's' to the left of JOBNAME with JES JOB number from above
   c) Observe content of JOB
   d) Locate JCL error in output
   e) Fix the JCL error in JCLERROR
   f) Submit JCLERROR to verify correction.

LAB 3
=====

SORT JCL PROCEDURE
==================

1. Edit and submit JCL member to JES for execution
   a) TSO/ISPF =3.4
   b) Enter ###### in the Dsname level field
   c) Enter 'e' to the left of ###### .JCL
   d) Enter 's' to the left of SORT member
   e) Enter submit
   f) Observe and note JES JOB number.

2. Review JCL JOB output in SDSF
   a) Enter =sd;owner ###### ;st
   b) Enter '?' to the left of JES job number noted above
   c) Enter 's' to the left of each DDNAME in the output spool
   e) Observe the sorted output present in the JES spool

3. Edit and submit JCL member to JES for execution
   a) TSO/ISPF =3.4
   b) Enter ###### in the Dsname level field
   c) Enter 'e' to the left of ###### .JCL
   d) Enter 's' to the left of SORTPROC member
   e) Enter submit
   f) Observe and note JES JOB number.

4. Review JCL JOB output in SDSF
   a) Enter =sd;owner ###### ;st
   b) Enter '?' to the left of JES job number noted above
   c) Enter 's' to the left of each DDNAME in the JES output spool
   d) Observe this is an example of in-stream JCL PROC execution
   e) Observe the sorted output is not in the JES output spool
   f) Where was the sorted output written?

LAB 4
=====

Language compile, link and execution
==========

1. Edit and submit JCL member to JES for execution
   a) TSO/ISPF =3.4
   b) Enter ###### in the Dsname level field
   c) Enter 'e' to the left of ###### .JCL
   d) Enter 's' to the left of CRUN member
   e) Enter submit
   f) Observe and note JES JOB number.

2. Review JCL JOB output in SDSF
   a) Enter =sd;owner ###### ;st
   b) Enter '?' to the left of JES job number noted above
   c) Enter 's' to the left of each DDNAME in the output spool
   e) Observe this compiled and executed a C program
   f) Observe the program wrote output to spool and to a data set
   g) Can identify the data set containing the C source code?

3. Edit and submit JCL member to JES for execution
   a) TSO/ISPF =3.4
   b) Enter ###### in the Dsname level field
   c) Enter 'e' to the left of ###### .JCL
   d) Enter 's' to the left of ASMRUN member
   e) Enter submit
   f) Observe and note JES JOB number.

4. Review JCL JOB output in SDSF
   a) Enter =sd;owner ###### ;st
   b) Enter '?' to the left of JES job number noted above
   c) Enter 's' to the left of each DDNAME in the output spool
   e) Observe this compiled and executed an assembler program
   f) Observe the program wrote output to spool and to a data set
   g) Can identify the data set containing the assembler source code?

5. Edit and submit JCL member to JES for execution
   a) TSO/ISPF =3.4
   b) Enter ###### in the Dsname level field
   c) Enter 'e' to the left of ###### .JCL
   d) Enter 's' to the left of COBRUN member
   e) Enter submit
   f) Observe and note JES JOB number.

6. Review JCL JOB output in SDSF
   a) Enter =sd;owner ###### ;st
   b) Enter '?' to the left of JES job number noted above

c) Enter 's' to the left of each DDNAME in the output spool
e) Observe this compiled and executed an COBOL program
f) Observe the program wrote output to spool and to a data set
g) Can identify the data set containing the COBOL source code?

LAB 5
=====

COBOL compile, link and execution
==========

You will be provided COBOL source code, program input data, and JCL to compile, link and execute the COBOL program in batch for 6 COBOL program executions.

These COBOL programs are built upon one another with slight coding advancements. Minor COBOL source errors will exist in the 5th and 6th COBOL programs which will need to be identified and corrected.

1. Compile, link, execute and review the output of first COBOL program
   **Remember to replace ###### with your assigned ###### .

   a) Submit '###### .JCL(CBL0001J)'

   b) Enter the following on the SDSF command input line:
      ===> owner ###### ;prefix *;st

   c) Place a ? to the left of CBL0001J and observe:
      ** 6 entries under the DDNAME column
      ** 2 entries under ProcStep column
      ** DDNAME SYSPRINT with ProcStep COBOL
      ** DDNAME SYSPRINT with ProcStep LKED
      ** DDNAME PRTLINE

   d) Review DDNAME PRTLINE
      Place an 's' to the left of PRTLINE DDNAME to select
      Observe the data records in PRTLINE DDNAME output
      F3 will exit back to the previous panel

   e) Review DDNAME SYSPRINT from ProcStep COBOL
      The bottom of the output includes information about the
      success or failure of the COBOL source code compile.
      When a compile fails, the information in this output is
      used to identify and resolved the source code problem.

   f) Review DDNAME SYSPRINT from ProcStep LKED
      The bottom of the output includes information about the
      the success or failure creating the COBOL executable module.
      When the link fails, the information in this output is
      used to identify and resolved the source code problem.

   g) Review DDNAME JESJCL
      The output provides Job Control Langauge (JCL)
      substitution. The substition includes symbolic

resolution and JCL statement overrides.

h) Review DDNAME JESMSGLG
JES2 job log provides system information about the this jobs start, execution and stop time.

i) Review DDNAME JESYSMSG
The output provides system message about the execution of the entire JCL JOB. It includes condition codes for each JOB execution step.

j) The compile execution SYSIN DD JCL statement is the location of COBOL source code. Find the data set name and data set member name of the COBOL source code.

k) Find the data set name on the ACCTREC DD statement which is used by the program to read data during program execution. This data set can be browsed because it is a sequential data set.

2. Create VSAM data set using sequential data set input
a) Edit '###### .JCL(DEFVSAM)'
b) Submit '###### .JCL(DEFVSAM)'
c) Review the job output in SDSF for success.
d) This output copies the program input sequential data records to Virtual Storage Access Method data set (VSAM).

3. Compile, Link and Execute second COBOL program in batch
a) Submit '###### .JCL(CBL0002J)'
b) Review the job output in SDSF.
c) What is present in PRTLINE output?
d) Review '###### .JCL(CBL0002J)'
e) The program input records are from ACCTREC DD statement.
f) What data set name is on the ACCTREC DD statement?
*Note: This is the VSAM data set create in 1B.
g) Can you find the COBOL syntax change that enabled this program to read VSAM data set?
h) Compare the COBOL source file control statements in this program to the CBL0001 COBOL source. You should now be able to answer the above question.

4. Compile, Link and Execute third COBOL program in batch
a) Submit '###### .JCL(CBL0003J)'
b) Review the job output in SDSF.
c) Observe this program create a different output in PRTLINE
d) Observe the difference between the second and third COBOL source

5. Compile, Link and Execute fourth COBOL program in batch
a) Submit '###### .JCL(CBL0004J)'

b) Review the job output in SDSF.
c) Observe this program create a different output in PRTLINE
d) Observe the difference between the third and fourth COBOL source

6. Compile, Link and Execute fifth COBOL program in batch
   a) Submit '###### .JCL(CBL0005J)'
   b) Review the job output in SDSF
   c) Locate and correct COBOL source compile problem
      **Review the COBOL compile output for messages
      **Use the CBL0004 COBOL source code as an aid
   d) Observe this program create a different output in PRTLINE
   e) Observe the difference between the fourth and fifth COBOL source

7. Compile, Link and Execute sixth COBOL program in batch
   a) Submit '###### .JCL(CBL0006J)'
   b) Review the job output in SDSF
   c) Locate and correct COBOL source compile problem
   d) Observe this program create a different output in PRTLINE
   e) Observe the difference between the fifth and sixth COBOL source

LAB 6
=====

This lab is for the ambitious eager to practice and learn more.

1. a) Submit '###### .JCL(LOG)'
   b) Review the JCL and output
   c) How could this be useful?

2. a) Submit '###### .JCL(LOG1)'
      ....some of the same with a slight variation
   b) Review the JCL and output
   c) How could this be useful?

3. a) Submit '###### .JCL(LOG2)'
      ....some of the same with a slight variation
   b) Review the JCL and output
   c) How could this be useful?

4. a) Submit '###### .JCL(LOG3)'
      ....some of the same with a slight variation
   b) Review the JCL and output
   c) How could this be useful?

5. a) Submit '###### .JCL(LOG4)'
      ....some of the same with a slight variation
   b) Review the JCL and output
   c) How could this be useful?

6. a) Submit '###### .JCL(SENDMSG)'
      ....different idea
   b) Review the JCL and output
   c) How could this be useful?

7. a) Submit '###### .JCL(NETRPT)'
      ....another idea
   b) Review the JCL and output
   c) How could this be useful?

LAB 7
=====

System Commands and Configuration
==========

1. Review system command output
   a) From ISPF enter =sd;ulog
   b) Display system processor information
      /D M=CPU
   c) Display system memory (frames)
      /D M=STOR
   d) Display system pagespace (slots)
      /D M=ASM
   e) Display number of TSO users
      /D TS,L
   f) Display number of active address spaces
      /D A,L
   g) Display system IPL and parameter information
      /D IPLINFO
   h) Display system parameter library concatenation
      /D PARMLIB
   i) Display system executable module search list (aka linklist)
      /D PROG,LNKLST
   j) Display Unix System Services mounted filesystems
      /D OMVS,F
   k) Display Unix System Services parameters
      /D OMVS,O
   l) Display Unix System Services tasks as seen by MVS
      /D OMVS,A=ALL
   m) Display disk storage information
      /DEVSERV QDASD,TYPE=all
   n) Display JES spool space
      /$D SPOOL
   o) Display JES automatic commands
      /$T A,ALL

   Feel free to try other MVS and JES2 Display commands
   located in the MVS and JES2 commands reference manual

2. Review system configuration information
   a) From ISPF enter TSO SYSLIB
      SYSLIB is a CLIST (command list) which allocates the
      the system parameter library concatentation and the
      system JCL procedure library concatentation, then
      an ISPF routine (ISRDDN) is executed which displays
      all the session data set allocations.
   b) Enter b to the left of $PARMLIB data set concatenation
      PDS members read by the system during system initialization

are displayed. Several IEASYSxx members are read during system initialization. The members read are known by D IPLINFO command output which contained:
IEASYS LIST = (00,LV,SV,VN)

c) Select (s) to browse IEASYS00
d) Select (s) to browse IEASYSLV
e) Select (s) to browse IEASYSVN
f) Select (s) to browse IEASYSSV
Observe entry:
CMD=(&JESSSN.,00,LV,LW,SV,VN),
This results in COMMNDxx members to be read
&JESSSN variable is J2
Observe entry:
MSTRJCL=SV,
This is the Master Scheduler JCL procedure.
g) Select (s) to browse COMMNDJ2
h) Select (s) to browse COMMND00
i) Select (s) to browse COMMNDLV
j) Select (s) to browse COMMNDLW
k) Select (s) to browse COMMNDSV
l) Select (s) to browse COMMNDVN
m) F3 again to return to data set allocation list
n) Enter b to the left of $PROCLIB data set concatenation
o) Select (s) to browse the JES2 JCL procedure which is started during system initialization via COMMNDJ2
Observe //PROC00 DD data set concatentation
These are the data sets searched for JCL procedures when JCL is submitted to JES for batch execution or when JCL is started as a foreground task.

z/OS Unix System Services Filesystems
==========

3. Edit and submit JCL member to JES for execution
a) TSO/ISPF =3.4
b) Enter ###### in the Dsname level field
c) Enter 'e' to the left of ###### .JCL
d) Enter 's' to the left of ZFS member
e) Change all occurrences of ### to your uniquely assigned number
f) Enter submit
g) Review output
If successful, then you just created a new z/OS unix system services filesystem for yourself.

LAB 8
=====

DB2 for z/OS
============

1. Use ISPF Interactive DB2 Interface SPUFI to create a DB2 object.
   a) ISPF =D2 to enter DB2 Interface panel
   b) Select D - DB2I DEFAULTS to set global parameters
   c) Enter DBCG in DB2 NAME field <enter> - then F3 to exit/return
      SSID: DBCG should appear in upper right corner of panel
   d) Select 1 - SPUFI to process SQL statements
   e) Provide input and output data set

      Enter the input data set name:
       1 DATA SET NAME ... ===> DB2.SQL(ACCT2)

      Enter the output data set name:
       4 DATA SET NAME ... ===> DB2.SQL.ACCT2.OUTPUT

   f) The next panel presents chance to modify attributes
      Accept defaults and just depress <enter>
   g) SQL statements are present in ISPF editor
      change all occurrences of ###### to your unique number
   g) <enter> F3 to save changes are return to SPUFI
   h) <enter> again to exit the SQL statements
   i) SQL statement execution output will be displayed
      Output should include:
       TABLESPACE created
       TABLE created
       GRANT given to public
       INDEX created

2. Load data into the DB2 TABLE created above
   a) ISPF =3.4 enter ######.DB2.JCL in the Dsname field
   b) Select e to the left of ######.DB2.JCL
   c) Select s ACCTLOAD member
   d) Change all occurrences of ###### to your unique number
   e) submit JCL
   f) Use SDSF to check job output for successful completion

3. Use ISPF Interactive DB2 Interface SPUFI to create a DB2 object.
   a) ISPF =D2 to enter DB2 Interface panel
   b) Select 1 - SPUFI to process SQL statements
   c) Provide input and output data set

      Enter the input data set name:
       1 DATA SET NAME ... ===> DB2.SQL(ACCT3)

Enter the output data set name:
4 DATA SET NAME ... ===> DB2.SQL.ACCT3.OUTPUT

d) The next panel presents chance to modify attributes
Accept defaults and just depress <enter>
e) SQL statements are present in ISPF editor
change all occurrences of ###### to your unique number
**Note: -- in position 1 and 2 makes statement a comment
remove -- to execute desired statement
add -- to make the statement a comment
f) <enter> F3 to save changes are return to SPUFI
g) <enter> again to exit the SQL statements
h) SQL statement execution output will be displayed

ISPF Editor – **Line** Command Summary

Basic line commands:

Insert	i	i#	
Delete	d	d#	dd
Repeat	r	r#	rr
Copy	c	c#	cc
Move	m	m#	mm
After	a		
Before	b		
Overlay	o	o#	oo

Advanced line commands:

Shift columns left	(	(#	((
Shift columns right	)	)#	))
Shift data left	<	<#	<<
Shift data right	>	>#	>>
Exclude	x	x#	xx
Show	x	s#	

First	f	f#
Last	l	l#

Text Handling line commands:

te	format screen for power typing text paragraphs
tf	flow text to the end of a paragraph
ts	split a text line at the cursor to allow insertion
lc	change text from uppercase to lowercase
uc	change text from lowercase to uppercase

Miscellaneous line commands:

bnds	display and allow changes to the current bounds
cols	display formatted line for identifying display columns
mask	display and allow changes to the current mask
tabs	display and allow changes to the current tabs line

Labels line ranges

.lab#1	chosen label
.lab#2	chosen label
.lab#3	chosen label

example - change ‘abc’ ‘123’ .lab#1 .lab#2 all

ISPF Editor – **Primary** Commands Summary

save	save the data without ending the edit session
cancel	cancel edit without saving the data
find	find a specified character string
change	find and change a specified character string
exclude	exclude lines that contain a specified character string
locate	display a particular line in the data
sort	sort by specified fields
create	create a new member of a partitioned data set
replace	replace a member or an entire sequential data set
copy	copy data from a member or sequential data set
move	move (copy and delete) a member or sequential data set
compare	display differences between data sets
submit	submit edit JCL as a job stream for background execution
profile	display the current edit profile
recovery	on/off
number	on/off
caps	on/off

hex	on/off
nulls	on/off
tabs	on/off
note	on/off
autonum	on/off
autolist	on/off
stats	on/off
autosave	on/off
pack	on/off
reset	reset the edit display
hilite	color code formatting
undo	

Data FTP Interface

put	*transmit data to server*
get	*retrieve data from server*
mget	*retrieve multiple data members from server*
mput	*transmit multiple data members to server*
prompt	*toggle off/on prompt*
cd 'mvs_name'	*change directory to mvs data set name or HLQ*
cd /	*change directory to unix filesystem path or subdirectory*
dir	*list members*

JES FTP Interface

quote site filetype=jes
quote site jesjobname=* jesstatus=all jesowner=*

put	*transmit JCL to JES reader*
get	*retrieve output from JES output spool*
dir	*list jes spool content*
quote site filetype=seq	*back to data interface mode*