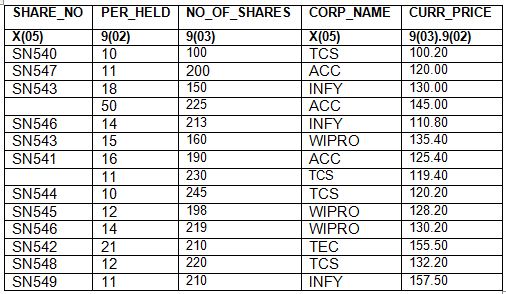
MINI CASE STUDY: **SHARE**

**STEP1:** **ISPF**

* Allocate a PS dataset with record length 80 with naming convention as below.  
  PS1 - <USERID>.L1J.SHARE.PS1
* Using the details from below file layout, enter records into the PS1 file as per the instructions given,
  + **Do not enter 1strow** in PS file. 1st row contains header details for reference.
  + **Do not enter 2nd row** in PS file. 2nd row contains layout details for reference.
  + One space filler had to be inserted between each field in the PS file.
  + All alphanumeric data to be entered in **CAPITAL** letters.

****

**STEP2: JCL**

1. JCL member naming convension **JA11J<yyy>,** where <yyy> denotes last 3 digits of your user ID. Proper JOB card without any **RESTART** step to be given.

* + Allocatea VSAM KSDS dataset with the following specifications,  
    RECORDSIZE (80, 80)  
    Name: <USERID>.L1J.SHARE.KSDS  
    Key Field: Share\_No X(05)

**STEP3:  JCL**

1. JCL member naming convention **JA21J<yyy>,** where <yyy> denotes last 3 digits of your user ID. Proper JOB card without any **RESTART** step to be given.

* + Using Sort utility perform the below operations on PS1 and store the output in PS2 file. PS2 -> <USERID>.L1J.SHARE.PS2. First step in this job should be the DELETE step for PS2 file.

i. Sort records in ascending order based on SHARE\_NO and NO\_OF\_SHARES.

ii. Eliminate records which have with SHARE\_NO as blanks.

iii. Remove duplicates for each SHARE\_NO.  
Ex : If Share No SN541 contains 3 records after sorting, write the first record alone.

iv. Extract 8 records from 2nd record into output.  
Ex: If file contains 15 records after sorting and eliminating blank SHARE\_NO, this step should extract 2nd to 9th records.

2. Split the PS2 file into two new files using Sort utility based on CORP\_NAME.The layout for the two new files will be the same as the PS2 file layout.  
The Job step have to be stored in **JA31J<yyy>**. First step should be the delete step for ACC and INFY files. Proper JOB card without any **RESTART** step to be given.

* + Write the records with CORP\_NAME= ‘ACC’ into <USERID>.L1J.SHARE.ACC
  + Write the records with CORP\_NAME= ‘INFY’ into <USERID>.L1J.SHARE.INFY
  + Combine the data of both the above recordsand store the output in PS3 file. PS3 -> <USERID>.L1J.SHARE.PS3. First step in this job should be the DELETE step for PS3 file.

Note : The merged output records should be in ascending order based on SHARE\_NO.

**STEP4: COBOL**

Input file to be used in the program **:** <USERID>.L1J.SHARE.PS2

**DD name to be used : INPSHRPS**

Output file to be used in the program **:** <USERID>.L1J.SHARE.KSDS

**DD name to be used : OUTSHRKS**

Output file to be used in the program **:** <USERID>.L1J.SHARE.ERR

**DD name to be used : ERRSHRPS**

*Note: Please use only the* **above mentioned DD names.**

Write a COBOL program to perform the following,

* + Read records from input PS2 dataset(<USERID>.L1J.SHARE.PS2) and validate input values for each field in the input file.

Check whether PER\_HELD is numeric.

Check whether NO\_OF\_SHARES is numeric.

Check whether CORP\_NAME field is not blank.

Check whether CURR\_PRICE is **numeric before and after decimal point**.

* + If the input record does not pass through the validations specified above successfully, then write that input record into error PS file <USERID>.L1B.SHARE.ERR and start processing the next record.   
    Layout of ERROR file is as below
    - * The first record must be a header record with required spacing between the fields as mentioned below.  
          
        https://assessment.cognizant.com/assessment/file.php?file=%2F324%2FLAB1_IJ_Nov14%2F1J_2.JPG

*Note : One space filler to be inserted between each field.*

ERR\_FIELD contains the error field name.

Ex:If the PER\_HELD field is not numeric, then the ERR\_FIELD should contain ‘PER-HELD’

If CORP\_NAME is blank, , then the ERR\_FIELD should contain ‘CORP-NAME’

* + Do the following processing for every record which has passed through the validations successfully.
    - Load the Share Number, No.of shares, corporate name and percentage held into COBOL one dimensional table.
    - From the table, find the records which have percentage held value greater than 10%, and do the following calculations.

1. Calculation of New Current price

* + - For the above records which has percentage value field greater than 10%, calculate New Current price using the below formula, and skip the other records from processing.  
        
      New Current Price = No: of shares + Square root (No :of shares

2. CLASS and RANGE Calculation

* + - * For shares with CORP\_NAME as INFY and PER\_HELD >15
        + Write CLASS field of corresponding record as ‘C1’ and RANGE as’ H
      * For shares with CORP\_NAME as ACC and PER\_HELD >15
        + Write CLASS field of corresponding record as ‘C2’ and RANGE as’ H
      * For shares with CORP\_NAME as TCS and PER\_HELD >10
        + Write CLASS field of corresponding record as ‘C3’ and RANGE as’ M
      * For all other shares,
        + Write CLASS field of corresponding record as ‘C4’ and RANGE as’ L’

3. Outstanding share price and income tax calculation

* + Using a sub-program, calculate (i) Outstanding share price and (ii) Income Tax using below formulas . Both the values should be rounded off to three decimal places.

a. Calculate Outstanding share price as

**OUT\_SHR\_PRICE=New Current price \* NO\_OF\_SHARES**

*Note : new Current price refers to price calculated in previous step*

b. Calculate INC\_TAX as

**INC\_TAX=(Divide New Current price INTO OUT\_SHR\_PRICE)\* PER\_HELD**

Return the values of OUT\_SHR\_PRICE and INC\_TAX to the main program to write to output file.

* + Write output record into KSDS file in the below format.

*NOTE :* **One space** *filler to be inserted between each field.*

https://assessment.cognizant.com/assessment/file.php?file=%2F324%2FLAB1_IJ_Nov14%2F1J_3.JPG

**Note:** *The* CURR\_PRICE *should be the new current price calculated as mentioned above (in Calculation No 1).*

* Compile and run the above COBOL program to achieve the results.

*JCL Member Naming Convention:* **JA41J<yyy>**, where <yyy> denotes the last 3 digits of your user ID. Proper JOB card without any **RESTART** step to be given.

**INSTRUCTIONS:**

* Follow the proper coding standards.
* Provide proper error handling routines.
* Place all the final deliverables into the PDS dataset :  **‘<USERID>.<X>.<Y>.PDS’ .**Where X ->  L1 and Y denotes 8 digit Batch name. Example:  Y- CHNMJ001  
                 [Example of **L1 PDS              – TECN001.L1.CHNMJ001.PDS]**                 
   For JCL’s, the member name should be “JA<x>1G<yyy>”  
  Note: Where <x> denotes the member no.(1 for 1st, 2 for 2ndmember) and <yyy> denotes the last 3 digits of your user ID.
  + - JCL's should be named in the member names as suggested in the respective steps.
* For COBOL Program, the member name should be “CA<x>1G<yyy>”  
  Note: Where <x> denotes the member no. (1 for 1st, 2 for 2ndmember) and <yyy> denotes the last 3 digits of your user ID.  
  If there are 2 members for a COBOL (Main program and sub program) created by ID TECN001, the member name should be “CA11G001” and”CA21G001”.

**EXPECTED DELIVERABLES in Mainframe PDS ‘<USERID>.<X>.<Y>.PDS’:**

* JCL for STEP2 and STEP3 in member names suggested in steps.
* COBOL programs in member names suggested.
* RUNJCL for the COBOL program.

.