Keep Statement:-

Specifies the variables to include in output SAS data sets.

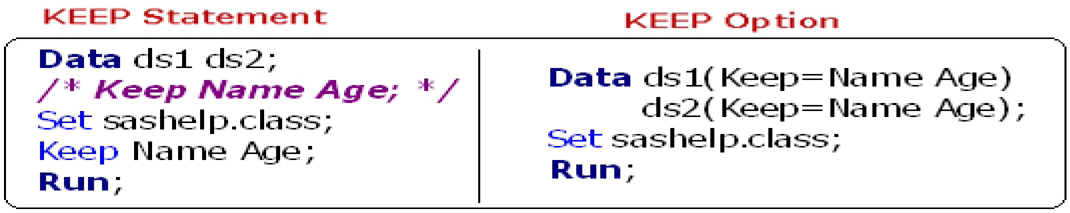
Syntax: KEEP variable-list;

Difference between KEEP Option and KEEP Statement

The **KEEP statement** applies to all SAS data sets that are created within the same DATA step and can appear anywhere in the step. The **KEEP Option** should specify for each dataset that are created within the same DATA step and appears as a dataset option.

The **KEEP statement** cannot be used in SAS PROC steps. The **KEEP= data set option** can use in SAS PROC steps.

The **KEEP statement** works before **PDV** but **KEEP option** works after **PDV**



KEEP Statement

```
Data ds1 ds2;
/* Keep Name Age; */
Set sashelp.class;
Keep Name Age;
Run;
```

KEEP Option

```
Data ds1(Keep=Name Age)
     ds2(Keep=Name Age);
Set sashelp.class;
Run;
```

Drop Statement:-

Excludes variables from output SAS data sets.

Syntax: DROP variable-list;

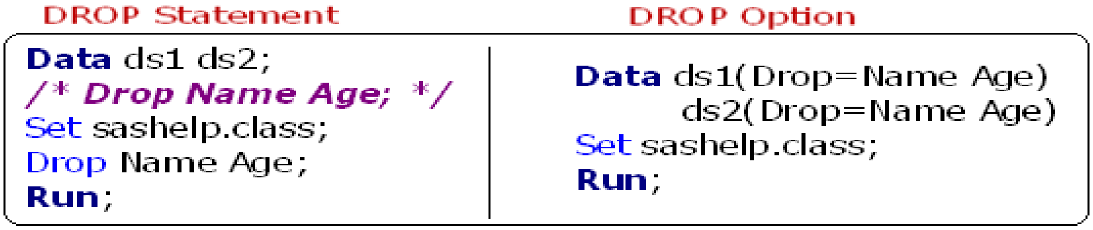
STANSYS SOFTWARE

Difference between DROP Option and DROP Statement

The DROP statement applies to all the SAS data sets that are created within the same DATA step and can appear anywhere in the step. The **DROP Option** should specify for each dataset that are created within the same DATA step and appears as a dataset option.

The **DROPstatement** cannot be used in SAS PROC steps. The **DROP= data set option** can use in SAS PROC steps.

The **DROPstatement** works before **PDV** but **DROP option** works after **PDV**



DROP Statement

```
Data ds1 ds2;
/* Drop Name Age; */
Set sashelp.class;
Drop Name Age;
Run;
```

DROP Option

```
Data ds1(Drop=Name Age)
     ds2(Drop=Name Age)
Set sashelp.class;
Run;
```

Rename Statement:-

Rename variables permanently in SAS datasets.

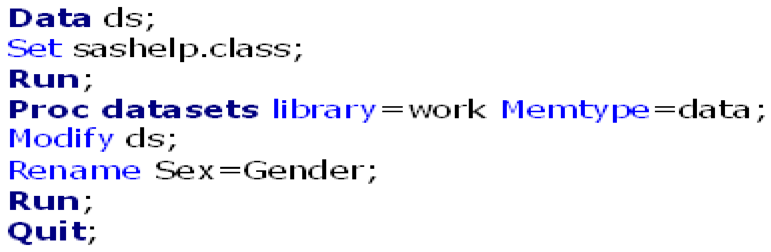
Syntax: RENAME Oldname=Newname;

Difference between RENAME Option and RENAME Statement

The **RENAME statement** applies to all the SAS data sets that are created within the same DATA step and can appear anywhere in the step. The **RENAME Option** should specify for each dataset that are created within the same DATA step and appears as a dataset option.

The **RENAME statement** works before **PDV** but **RENAME option** works after **PDV**

The **RENAME statement** cannot be used in SAS PROC steps. But Rename statement we can use in Proc datasets with Modify statement



```
Data ds;
Set sashelp.class;
Run;
Proc datasets library=work Memtype=data;
Modify ds;
Rename Sex=Gender;
Run;
Quit;
```

Where Statement:-

Selects observations from SAS dataset when it meet condition

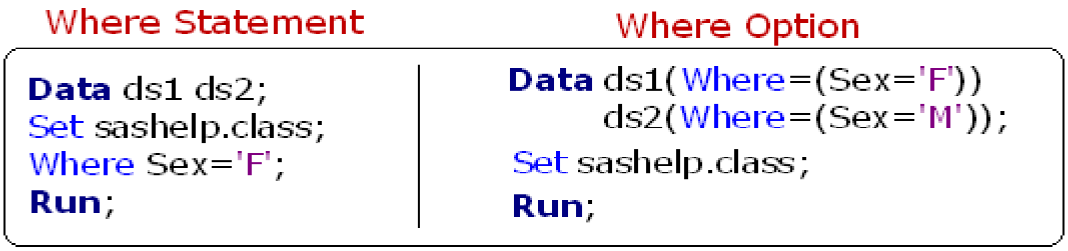
Syntax: WHERE Variable=Condition;

Difference between WHERE Option and WHEREP Statement

The **WHERE** statement applies to all the SAS data sets that are created within the same DATA step. The **WHERE Option** should specify for each dataset that are created within the same DATA step and appears as a dataset option.

The **WHERE statement** can be used in both DATA & PROC steps.

The **WHERE statement** works before **PDV** but **WHERE option** works after **PDV**



Where Statement

```
Data ds1 ds2;
Set sashelp.class;
Where Sex='F';
Run;
```

Where Option

```
Data ds1(Where=(Sex='F'))
     ds2(Where=(Sex='M'));
Set sashelp.class;
Run;
```

Label Statement:-

Assigns descriptive labels to variables.

Assign alias names to variables.

Label can’t change the column name permanently like rename option.

Syntax: - LABEL variable-1='label-1' . . . <variable-n='label-n'>;
LABEL variable-1=' ' . . . <variable-n=' '>;

```
Data DS1;
Infile datalines;
Input id name$ sex$ age sal ;
Label name='Emp Name' sex='Gender' sal='Income';
Datalines;
001 Ronald m 23 50000
002 Clark f 22 34500
003 Roopa f 26 45000
;
Run;
```

Remove Statement:-

Deletes an observation from a SAS data set.

Syntax: - REMOVE <data-set-name(s)>;

Remove statement works only with a MODIFY statement.

```
Data ds;
Set sashelp.class;
Run;
Data ds;
Modify ds;
If age=12 then remove;
Run;
```

```
Data ds1 ds2 ds3;
Set sashelp.class;
Run;
Data ds1 ds2 ds3;
Modify ds1;
Modify ds2;
Modify ds3;
If age=15 then remove;
Run;
```

```
Data ds1 ds2 ds3;
Set sashelp.class;
Run;
Data ds1 ds2 ds3;
Modify ds1;
Modify ds2;
Modify ds3;
If age=12 then remove ds1;
If age=15 then remove ds2;
If age=16 then remove ds3;
Run;
```

Delete Statement:-

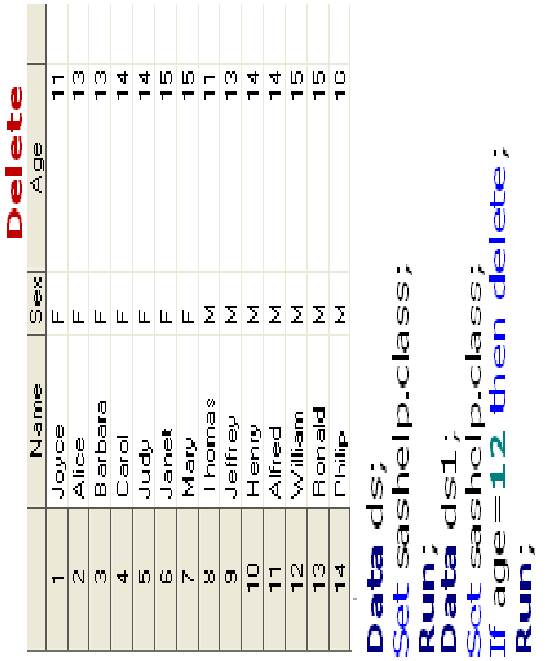
Stops processing the current observation.

Syntax: - DELETE;

When DELETE executes, the current observation is not written to a data set, and SAS returns immediately to the beginning of the DATA step for the next iteration.

```
Data ds1;
Set sashelp.class;
If age=12 then delete;
Run;
```

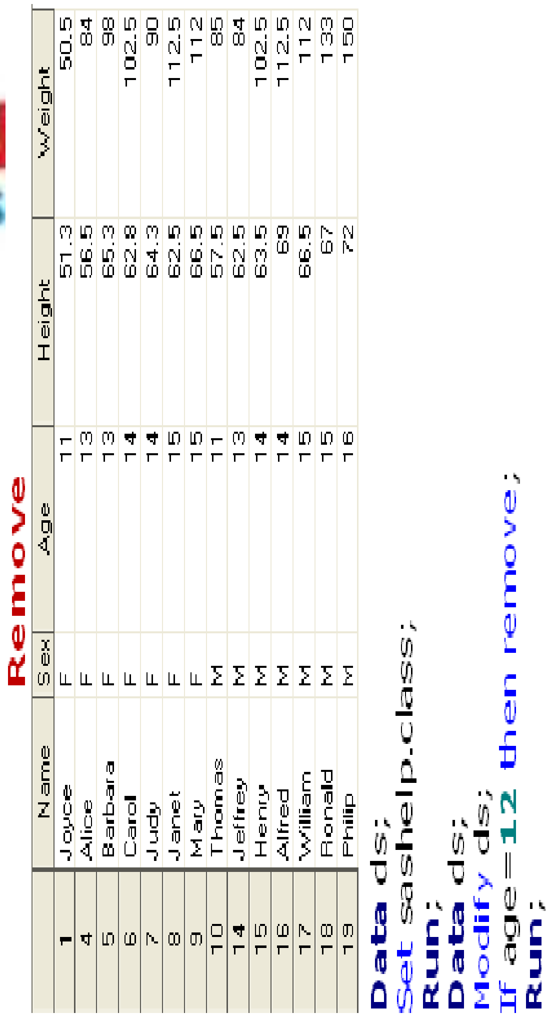
Observe observation numbers in right side diagram.



Delete

	Name	Sex	Age	
1	Joyce	F	11	
2	Alice	F	13	
3	Barbara	F	13	
4	Carol	F	14	
5	Judy	F	14	
6	Janet	F	15	
7	Mary	F	15	
8	Thomas	M	11	
9	Jeffrey	M	13	
10	Henry	M	14	
11	Alfred	M	14	
12	William	M	15	
13	Ronald	M	15	
14	Philip	M	16	

```
Data ds;
Set sashelp.class;
Run;
Data ds1;
Set sashelp.class;
If age=12 then delete;
Run;
```



Remove

	Name	Sex	Age	Height	Weight
1	Joyce	F	11	51.3	50.5
4	Alice	F	13	56.5	84
5	Barbara	F	13	65.3	98
6	Carol	F	14	62.8	102.5
7	Judy	F	14	64.3	90
8	Janet	F	15	62.5	112.5
9	Mary	F	15	66.5	112
10	Thomas	M	11	57.5	85
14	Jeffrey	M	13	62.5	84
15	Henry	M	14	63.5	102.5
16	Alfred	M	14	69	112.5
17	William	M	15	66.5	112
18	Ronald	M	15	67	133
19	Philip	M	16	72	150

```
Data ds;
Set sashelp.class;
Run;
Data ds;
Modify ds;
If age=12 then remove;
Run;
```

DM Statement:-

Syntax: - DM <window> 'command(s)' <window> <CONTINUE>;

Window: specifies the active window (like Editor, Log, Output)

Command(s): can be any windowing command or text editor command and must be enclosed in single quotation marks. If you want to issue several commands, separate them with semicolons.

Continue: causes SAS to execute any SAS statements that follow the DM statement in the Program Editor window and, if a windowing command in the DM statement called a window, makes that window active.

1) Clear Editor window.

```
DM 'Editor' Clear;
```

2) Clear Log window .

```
DM 'Log' Clear;
```

3) Clear Output window .

```
DM 'Output' Clear;
```

4) Clear both Log and Output window.

```
DM 'Log; Clear; Output; Clear;';
```

5) Saves the program.

```
DM 'flsvlast';
```

STANSYS
SOFTWARE SOLUTIONS

6) Closes the open dataset.

```
DM 'next viewtable:; end;';
```

7) If you want to clear the log and return back to the editor window after submitting DM statement.

```
DM Log 'clear;';
```

8) If you want to remain at Log window.

```
DM Log 'clear;' continue;
```

9) If you want to clear the log and output both, and then return to log window,

```
Dm 'log; clear; output; clear;' log;
```

10) Shows all the filenames.

```
DM 'Libname';
```

11) Shows all the filenames.

```
DM 'Filename';
```

12) Opens the last created dataset.

```
DM "vt &syslast";
```

13) If you want to see the column names and not the column labels.

```
DM "VT libname.dataset COLHEADING=NAMES" continue;
DM "VT work.ds1 COLHEADING=NAMES" continue;
```

14) To get properties window for changing attributes of variables.

```
DM 'var libname.dataset;' continue;
DM 'var work.ds1;' continue;
```

15) To clear a cluttered results window:

```
DM 'odsresults' clear ;
```

16) Export SAS datasets into Excel or Notepads.

```
DM "DEXPORT sashelp.class 'krishna.xls' replace";
DM "DEXPORT sashelp.class 'krishna.txt' replace";
```

```
NOTE: File "C:\Documents and Settings\Administrator\sample.xls" will be created if the
export process succeeds.
NOTE: "CLASS" range/sheet was successfully created.
```

create multiple sheets if same excel is referred multiple times with different datasets;

```
DM "DEXPORT sashelp.class 'krishna.xls' replace";
DM "DEXPORT sashelp.cars 'krishna.xls' replace";
```

17) Export LOG into specified file.

```
DM log 'E:\SAS\STANSYS\SAMPLE.TXT" ';
```

18) Export OUTPUT into specified file.

```
DM out  'E:\SAS\STANSYS\SAMPLE.TXT" ';
```

19) you wish to stop your program from going in an infinite loop, or during debugging you don’t want to run the whole code but a selection

```
DATA DS1;
A=2;
B=3;
RUN;
DM "WATTENTION";
DATA DS2;
A=5;
B=6;
RUN;
```

STANSYS SOFTWARE SOLUTIONS

20) If you want to close any window

```
DM wedit 'winclose';
DM log 'winclose';
DM out 'winclose';
```

21) To copy text

```
DM 'whostedit; include "filepath.filename.extension"; EDCMD selectall; EDCMD
     copy;  EDCMD winclose; ';
DM 'whostedit; include "E:\SAS\STANSYS\SAMPLE.TXT"; EDCMD selectall; EDCMD
     copy; EDCMD winclose; ';
```

SUM STATEMENT

Adds the result of an expression to an accumulator variable

Syntax: - variable+expression;

Examples:-

```
Data summ1;
a=2;
b=3;
c=4;
d=5;
Run;
```

```
Data summ2;
Set summ1;
Total=a+b+c+d;
Run;
```

```
Data summ3;
a=2;
b=3;
c=4;
d=.;
e=5;
f=.;
Run;
```

```
Data summ4;
Set summ3;
Total=a+b+c+d+e+g;
Run;
```

STANSYS SOFTWARE SOLUTIONS

Sum Function

```
Data summ5;
Set summ3;
Total=Sum(a,b,c,d,e,f);
Run;
```

Difference between SUM Statement and SUM Function

SUM statement works with only non missing values but SUM Function works with Missing values also.

SUM Statement Adds the value into variable with non missing values
If missing value are there after sum value is missing.
Ex: - x2=sum (4+.+9+3+8+.);
Above example gives value missing

SUM Function returns the sum of non missing values
Ex: - x2=sum (4,.,9,3,8,.); It gives value 24

```
Data Loan_info;
Infile datalines dsd;
Input Loan_id$ Cust_Name : $15.  Age   Loan_amt1:dollar5.
        Loan_amt2:dollar5. Loan_amt3 dollar5.;
Format Loan_amt1 dollar5. Loan_amt2 dollar5. Loan_amt3 dollar5.;
Datalines;
LP101,Ravi Sinha,23,$3000,$3500,$2000
LP102,Alan Nance,29,$2500,$1500,
LP103,Brown lee,31,$5000,$1000,$2000
LP104,Ashley McKnight,22,$1500, ,$3000
LP105,Jim Brown,25,$4500,$1000,$1200
;
Run;
```

```
Data Loan_info;
Set Loan_info;
Format Total1 dollar6. Total2 dollar6. ;
Total1=Loan_amt1+Loan_amt2+Loan_amt3; /* Sum Statement */
Total2=Sum(Loan_amt1,Loan_amt2,Loan_amt3); /* Sum Function */
Run;
```

RETAIN STATEMENT:

Retain the values of the variable in subsequent iterations of the data step.

Retain statement prevents SAS form re-initializing the values of new variables at the top of data step and can be used to create an accumulator variable.

Retain is useful to find out cumulative totals.

STANSYS
SOFTWARE SOLUTIONS

Syntax: - RETAIN <element-list(s) <initial-value(s)

Examples:-

```
Data Ret1;
Input Id Prod$ Sales;
Datalines;
101 Pen 230
102 Pencil 320
103 Book 210
104 Pen 210
105 Book 180
106 Pencil 310
;
Run;
```

```
Data Ret2;
Set Ret1;
Retain total 0;
Total=Total+Sales;
/*Total+Sales;*/
Run;
```

```
Data Ret2a;
Set Ret1;
Retain total ;
/*Total=Total+Sales;*/
Total=sum(Total, Sales);
Run;
```

Creating Rowno using Retain Statement

```
Data Ret3;
Retain Row;
Set Ret1;
Row+1;
Run;
```

Cumulative totals on each by group

```
Data ds;
Set sashelp.class;
Run;
Proc sort data=ds;
By sex;
Run;
```

----Way 1-------

```
Data ds2;
Set ds;
By sex;
Retain cum_ht 0;
If first.sex=1 then cum_ht=height;
Else cum_ht+height;
Run;
```

STANSYS
SOFTWARE SOLUTIONS

----Way 2-------

```
Data ds3(DROP=Y);
Set ds;
By sex;
Retain x 0 y 0;
If sex='F' then x=x+height;
Else y=y+height;
If y > 0 then x=y;
Run;
```

----Way 3-------

```
Data ds4;
Set ds;
By sex;
Retain x 0;
x=x+height;
If first.sex=1 then do;
x=height+0;
End;
Run;
```

Finding total sales for each EMPID wise

```
Data test;
Infile datalines;
Input empid  sales;
Cards;
1 500
2 890
9 500
1 650
3 740
2 600
4 700
5 320
9 360
3 450
;
Run;

Proc sort data=test out=test1;
By id;
Run;

Data result(drop=sales);
Set test1;
By id;
Retain total_sales;
If first.id then total_sales=0;
total_sales+sales;
If last.id then output;
Run;

Proc print data=result NOOBS;
Run;
```

STANSYS
SOFTWARE SOLUTIONS

OUTPUT STATEMENT

Writes the current observation to a SAS data set.

Syntax: - OUTPUT<*data-set-name(s)*>;

Example1:-

```
Data ds;
Set sashelp.cars;
If make='BMW' then output;
Run;
```

Example2:-

```
Data ds1 ds2;
Set sashelp.cars;
If make='BMW' then output ds1;
Else output ds2;
Run;
```

Example3:-

```
Data dsn;
Infile datalines;
Input a1 - a3;
Datalines;
1 2 3
4 5 6
2 3 4
3 4 5
7 8 9
;
Run;
Data ds(drop=a1-a3);
Set dsn;
a=a1;
output;
a=a2;
output;
a=a3;
output;
Run;
```

Example4:-

```
Data ds3;
Slice=2;
Set sashelp.class point=slice;
Output;
Stop;
Run;
```

STOP STATEMENT

Stops execution of the current Data step.

Syntax: - STOP;

Examples:-

```
Data ds1;
Infile datalines;
Stop;
Input idno name $ team $ strtwght endwght;
Cards;
1023 David red 189 165
1049 Amelia yellow 145 124
1219 Alan red 210 192
1246 Ravi yellow 194 177
1078 Ashley red 127 118
1221 Jim yellow 220 .
;
Run;
```

It create 0 observations dataset

Above data set WORK.DS1 has 0 observations and 5 variables.

```
Data DS2;
Stop;
Set sashelp.class;
Run;
```

STANSYS SOFTWARE SOLUTIONS

Above data set WORK.DS2 has 0 observations and 5 variables.

In above programs when input/set statement executes data should read into input buffer but we are using stop statement so it won’t read into input buffer, so pdv can’t assign data to variables that’s why we are getting 0 observations dataset.

See How STOP works with below examples

```
Data ds3;
Slice=2;
Set sashelp.class point=slice;
Output;
Stop;
Run;
```

In above program when it’s read 2nd observation it’s give into output and it stop to read other observations so it reduce execution time.

```
Data ds4;
Do Slice=2,5;
Set sashelp.class point=slice;
Output;
end;
Stop;
Run;
```

In above program when it's read 2nd and 5th observation it's give into output and it stop to read other observations so it reduce execution time.

```
Data ds5;
Do Slice=2,5;
Set sashelp.class point=slice;
Output;
Stop;
end;
Run;
```

In above program when it's read 2nd observation it's give into output and it stop to read other observations.

Observe below programs how the result is changing because of STOP Statement.

```
Data ds1;
Do i=1 to 10;
End;
Run;
```

```
Data ds2;
Do i=1 to 10;
Output;
End;
Stop;
Run;
```

STANSYS SOFTWARE SOLUTIONS

In above program when it's happened first loop I=1 and it takes into output dataset. Then it goes to end statement but loop is not ending so it goes second loop so I=2 and it takes into output dataset. Then it goes to end statement but loop is not ending so it goes third loop like this after all 10 loops stop statement works so I value is 1 to 10.

```
Data ds3;
Do i=1 to 10;
Output;
Stop;
End;
Run;
```

In above program when it's happened first loop I=1 and it takes into output dataset. Then it goes STOP statement before END statement. So it stop to read the data so I=1 only.

RETURN STATEMENT

Stops executing statements at the current point in the DATA step and returns to a predetermined point in the step.

Syntax: - Return

```
Data survey;
Input x y;
If x=y then return;
Put x= y=;
Datalines;
1 2
3 4
5 5
6 7
9 9
;
Run;
```

In above example, when the values of X and Y are the same, SAS executes the RETURN statement and adds the observation to the data set. When the values of X and Y are not equal, SAS executes the remaining statements and then adds the observation to the data set.

REPLACE STATEMENT

Replaces an observation in the same location.

Syntax: - REPLACE <data-set-name-1><. . .data-set-name-n>;

STANSYS SOFTWARE SOLUTIONS

```
Data master;
Input Name$ id$ Phone;
Datalines;
Kevin ABCjkh 904
Sandi defsns 905
Terry ghitDP 951
Jason jklJWM 962
;
Run;
Data trans;
Input Name$ id$ Phone;
Datalines;
. ABCjkh 2904
. defsns 2905
Madeline mnombt 2983
;
Run;
Data master;
Modify master trans;
By id;
/* obs found in master  */
/* change info, replace */
If _iorc_ = %sysrc(_sok) then replace;
/* obs not in master    */
Else if _iorc_ = %sysrc(_dsenmr) then do;
/* reset _error_        */
_error_=0;
/* reset _iorc_         */
_iorc_=0;
/* output obs to master */
Output;
End;
Run;
```

STANSYS
SOFTWARE SOLUTIONS

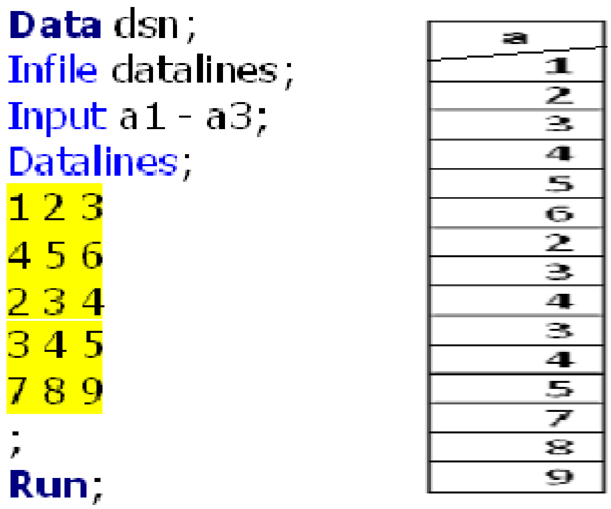
STANSYS
SOFTWARE SOLUTIONS

STANSYS
SOFTWARE SOLUTIONS

STANSYS
SOFTWARE SOLUTIONS

Interview Questions

Q1) What is difference between STATEMENT & OPTION?
Q2) What is difference between KEEP Statement & KEEP Option?
Q3) What is difference between DROP Statement & DROP Option?
Q4) What is difference between RENAME Statement & RENAME Option?
Q5) What is difference between WHERE Statement & WHERE Option?
Q6) What is difference between LABEL Statement & LABEL Option?
Q7) What is difference between RENAME Statement & LABEL Statement?
Q8) What is difference between SUM Statement & SUM Function?
Q9) What is difference between RETAIN Statement & SUM Statement?
Q10) What is difference between DELETE Statement & REMOVE Statement?
Q11) What is the significance of LABEL NAME=’Emp Name’ & LABEL NAME=’ ‘;
Q12) What is difference between DROP Statement & REMOVE Statement?
Q13) What is RETURN Statement ? Syntax? Example ?
Q14) What is REPLACE Statement ? Syntax? Example ?
Q15) What is STOP Statement ? Syntax? Example ?
Q16) How can you create 0(Zero) observations dataset? Syntax?
Q17) What is GOTO Statement ? Syntax? Example ?
Q18) What is OUTPUT Statement? Syntax? Example ?
Q19) Write code to get below output ? Using DSN data?



```
Data dsn;
Infile datalines;
Input a1 - a3;
Datalines;
1 2 3
4 5 6
2 3 4
3 4 5
7 8 9
;
Run;
```

a
1
2
3
4
5
6
2
3
4
3
4
5
7
8
9

ANSYS
ARE SOLUTIONS

Q20) What is X value? X=2+3+4+5;
Q21) What is X value? X=2+3+.+5;
Q22) What is X value? X=sum(2,3,4,5);
Q23) What is X value? X=sum(2,3,.,5);
Q24) What is the significance of below codes:

```
Data d1;     Data d2;     Data d3;     Data d4;      Data d5;
a=2;         a=2;         a=2;         a=2;          a=2;
b=3;         b=3;         b=.;         b=.;          b=.;
a+b;         c=a+b;       a+b;         c=a+b;        c=sum(a,b);
Run;         Run;         Run;         Run;          Run;
```

Q25) Why should I write length statement before Input statement?
Why not after Input statement?

Q26) How you can U create ROWNO/ID Variable using retain statement?

Q27) How can you select 2,5 observations from SASHELP.CLASS dataset?

Q28) Write code to find out total cumulative height for females and total cumulative height for males from SASHELP.CLASS dataset?

Q29) Write code to find out total height for each age group wise from SASHELP.CLASS? (Each age group should contain only one record with total height)

Q30) How can U export LOG into external files? Using DM statement?

Q31) How can U export OUTPUT into external files? Using DM statement?

Q32) How can U export SAS DATSETS into EXCEL/NOTEPADS using DM statement?

Q33) How can U delete observations from SAS Dataset ? Code it ?

Q34) Write outputs for below codes?

```
i) Data ds1;
    Do i=1 to 10;
    End;
    Run;

ii) Data ds2;
    Do i=1 to 10;
    Output;
    End;
    Stop;
    Run;
iii) Data ds3;
    Do i=1 to 10;
    Output;
    Stop;
    End;
    Run;
```

STANSYS
SOFTWARE SOLUTIONS

Q35) Data is available in the location of "E:\SAS\RAW_DATA\OTHER_STATEMENTS" Tasks are available at TASK notepad in above same location? Read that data into SAS and write the program for each task ?