

Dated: 15 April, 2018

```
.....  
.....  
  
.....
```

Do Statement: This is used for executing multiple statements for a condition. Always use end “with” do “statement”.

Syntax: if (condition) then do;

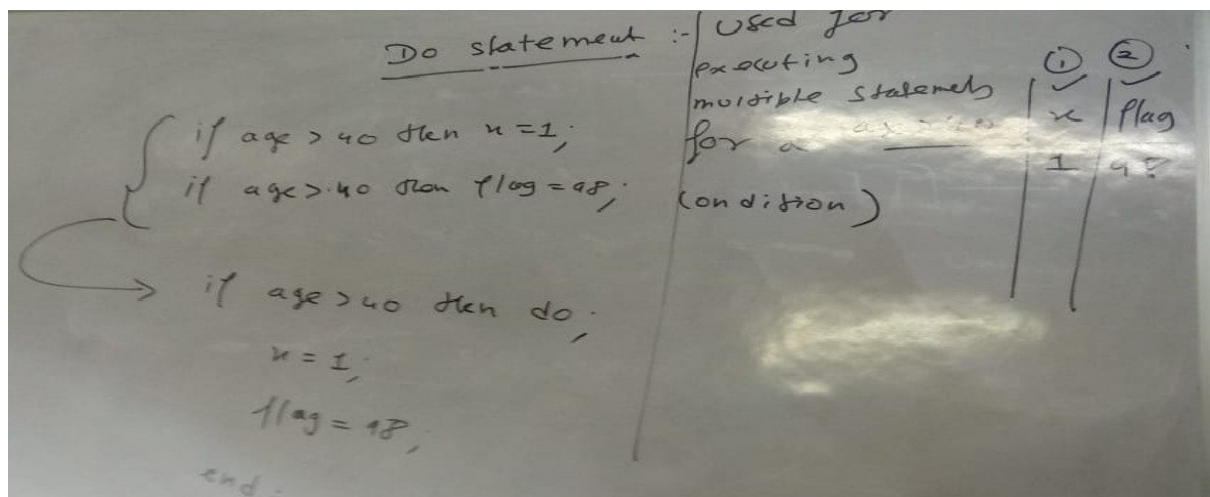
Statement 1;

Statement 2;

\\ and likewise other statements

end;

Note: by default increment in a Do Loop is by 1 only.



Example 1:

Q: If age > 40 then x = 1 and if age > 40 then flag = 98 (required).

```
if age > 40 then do;
```

```
x = 1;
```

```
flag = 98;
```

end;

Explained: Here when the condition of `age > 40` is satisfied, then the loop always executes the other 2 statements i.e- `x=1` & `flag = 98`.

```

data a;
set sasuser.admit;
if age gt 40 then do;
x=1;
flag=99;
end;
else do;
x=0;
flag=98;
end;
run;

data a;
set sasuser.admit;
if sex="M" then do;
gender="Male";
feel=fee+10;
end;
else do;
gender="female";
feel=fee-10;
end;
run;

```

Example 2:

```

data a;
set sasuser.admit;
if age gt 40 then do;
x =1;
flag = 99;
end;
else do;
x=0;
flag = 98;
end;
run;

```

Explained: Here when the condition of age > 40 is satisfied, then the do loop always executes x=1 & flag = 99 and if not satisfied then x=0 & flag = 98 is executed as output.

Example 3:

```

data a;
set sasuser.admit;
if sex = "M" then do;
gender = "Male";
feel = fee +10;
end;
else do;
gender = "female";
feel = fee-10;
end;
run;

```

Explained: Here when the condition of sex = "M" is satisfied, then the do loop always executes gender = "Male" and else gender = "female" is executed as output.

```

data a;
do i=1 to 5;

end;
run;data a;
do i=1 to 5;
output;
end;
run;
data a;
do i=1 to 5;
output;
end;
output;
run;

data a;
do i=1 to 5;
output;
output;
end;
output;
run;

```

Example 4:

data a;

Output

i

```
do i=1 to 5;                                6
```

```
end;
```

```
run;
```

Explained: here, i will increase from 1 to 5 but due to absence of explicit output statement within the loop, no output is displayed but when i=6 do loop fails and ends there now value for i=6 is displayed as implicit output because of default functionality of datastep

Example 5:

Output

```
data a;                                     i
do i=1 to 5;                               1
output;                                    2
end;                                        3
run;                                        4
                                           5
```

Explained: : here, i will increase from 1 to 5 but due to absence of explicit output statement within the loop, output is displayed as i=1 to 5 but for i = 6, do loop fails and ends there.

Example 6:

Output

```
data a;                                     i
do i=1 to 5;                               1
output;                                    2
end;                                        3
output;                                    4
run;                                        5
                                           6
```

Explained: here, i will increase from 1 to 5 due to the do loop and so the output as well, and 6 is displayed as output because of the final explicitly mentioned output statement.

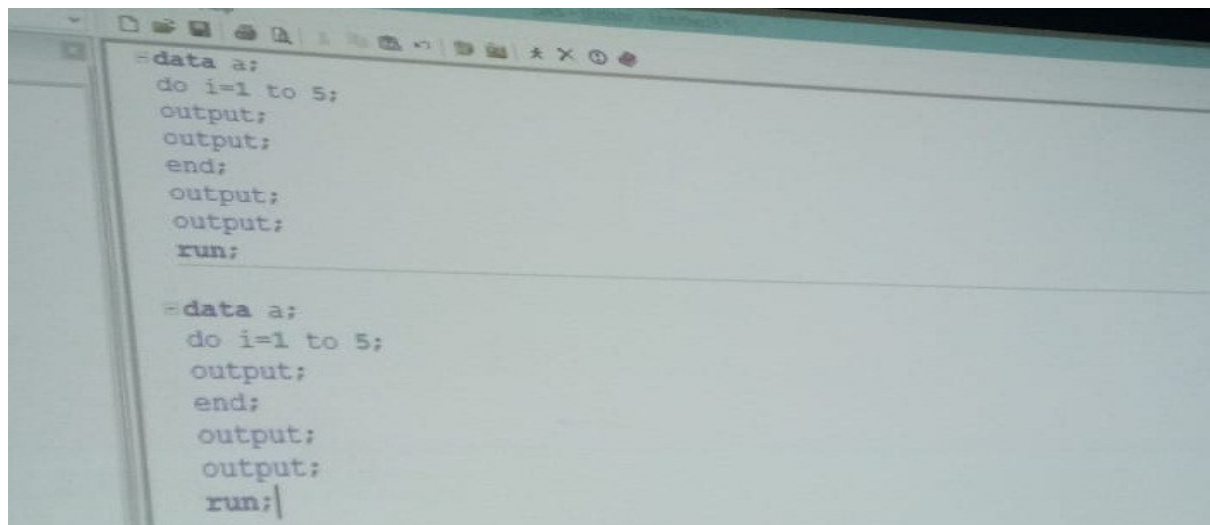
Example 7:

Output

```
data a;                                     i
do i=1 to 5;                               1
output;                                    1
output;                                    2
end;                                        2
output;                                    3
run;                                        3
```

4
4
5
5
6

Explained: for every value of i in the loop, output is encountered twice therefore printed twice in the output. Third output prints iterated value i=6 for which loop fails but output is outside loop.

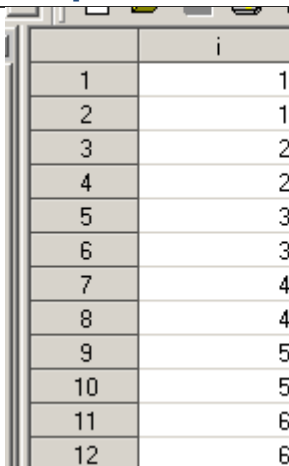


```
=data a;  
do i=1 to 5;  
output;  
output;  
end;  
output;  
output;  
run;  
  
=data a;  
do i=1 to 5;  
output;  
end;  
output;  
output;  
run;
```

Example 8:

```
data a;  
do i=1 to 5;  
output;  
output;  
end;  
output;  
output;  
run;
```

Output



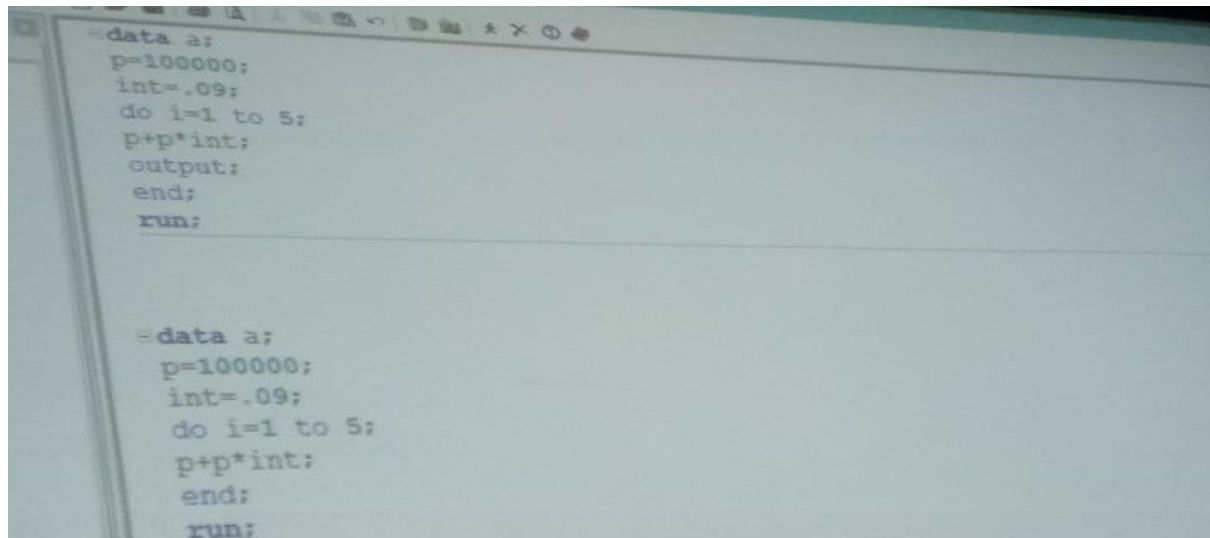
	i
1	1
2	1
3	2
4	2
5	3
6	3
7	4
8	4
9	5
10	5
11	6
12	6

Example 9:

Output

```
data a;
do i=1 to 5;
output;
end;
output;
output;
run;
```

	i
1	1
2	2
3	3
4	4
5	5
6	6
7	6



Example 10:

```
data a;
p=100000;
int=.09;
do i=1 to 5;
p+p*int;
output;
end;
run;
```

Output

	p	int	i
1	109000	0.09	1
2	118810	0.09	2
3	129502.9	0.09	3
4	141158.161	0.09	4
5	153862.39549	0.09	5

Example 11:

```
data a;
p=100000;
int=.09;
do i=1 to 5;
p+p*int;
end;
run;
```

Output

	p	int	i
1	153862.39549	0.09	6

```

= data a;
p=100000;
int=.09;
do i=1 to 5;
year+1;
p+p*int;
output;
end;
drop i;
run;

= data a;
p=100000;
int=.09;
do year=1 to 5;
p+p*int;
output;
end;
run;

```

<u>Example 12:</u>	<u>Example 13:</u>	<u>Output</u>																								
data a; p=100000; int=.09; do i=1 to 5; year+1; p+p*int; output; end; drop i; run;	data a; p=100000; int=.09; do year=1 to 5; p+p*int; output; end; run;	<table><tr><th></th><th>p</th><th>int</th><th>year</th></tr><tr><td>1</td><td>109000</td><td>0.09</td><td>1</td></tr><tr><td>2</td><td>118810</td><td>0.09</td><td>2</td></tr><tr><td>3</td><td>129502.9</td><td>0.09</td><td>3</td></tr><tr><td>4</td><td>141158.161</td><td>0.09</td><td>4</td></tr><tr><td>5</td><td>153862.39549</td><td>0.09</td><td>5</td></tr></table> <p>Explained: both ex 12, 13 gives same output, just the difference of using the increment value “ i ” in ex 12 and “ year ” in ex 13 for simplicity of code.</p>		p	int	year	1	109000	0.09	1	2	118810	0.09	2	3	129502.9	0.09	3	4	141158.161	0.09	4	5	153862.39549	0.09	5
	p	int	year																							
1	109000	0.09	1																							
2	118810	0.09	2																							
3	129502.9	0.09	3																							
4	141158.161	0.09	4																							
5	153862.39549	0.09	5																							

```

= data a;
p1=100000;
p2=100000;
int1=.09;
int2=.08;
do i=1 to 5;
p1+p1*int1;
p2+p2*int2;
loss=p2-p1;
output;
end;
run;

= data a;
p1=100000;
p2=100000;
int1=.09;
int2=.08;
do i=1 to 5;
p1+p1*int1;
p2+p2*int2;
end;
loss=p2-p1;
run;

```

<u>Example 14:</u>	<u>Example 15:</u>	<u>Output</u>
--------------------	--------------------	---------------

```
data a;
p1=10000
0;
p2=10000
0;
int1=.09;
int2=.08
do i=1 to
5;
p1+p1*int
1;
p2
+p2*int2;
loss=p2-
p1;
output;
end;
run;
```

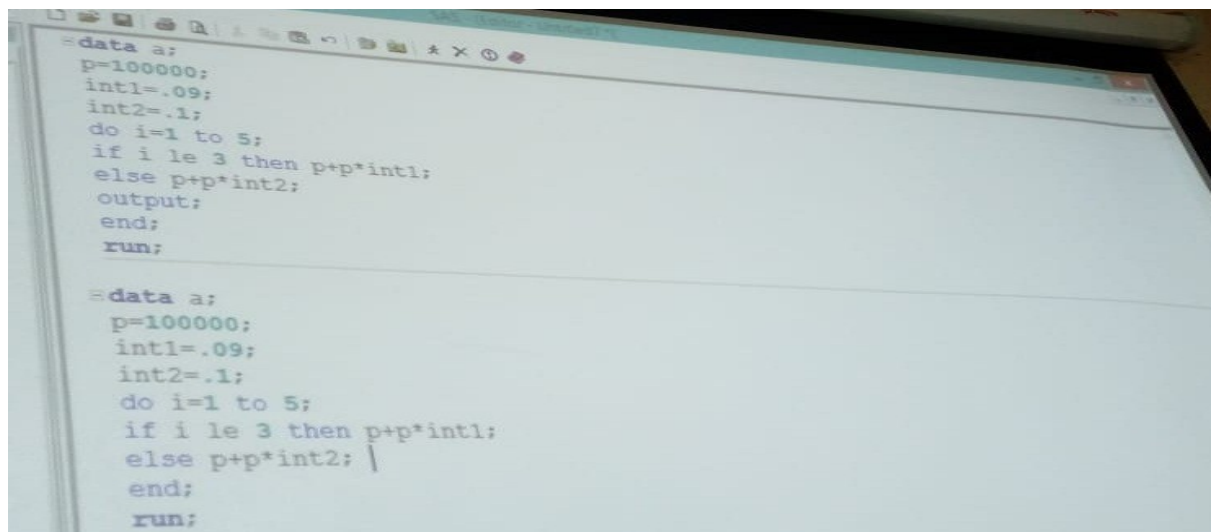
```
data a;
p1=10000
0;
p2=10000
0;
int1=.09;
int2=.08
do i=1 to
5;
p1+p1*int
1;
p2
+p2*int2;
end;
loss=p2-
p1;
run;
```

	p1	p2	int1	int2	i	loss
1	109000	108000	0.09	0.08	1	-1000
2	118810	116640	0.09	0.08	2	-2170
3	129502.9	125971.2	0.09	0.08	3	-3531.7
4	141158.161	136048.896	0.09	0.08	4	-5109.265
5	153862.39549	146932.80768	0.09	0.08	5	-6929.58781

Ex: 14—output displays loss in every observation because of loss statement within the loop boundary

	p1	p2	int1	int2	i	loss
1	153862.39549	146932.80768	0.09	0.08	6	-6929.58781

Ex: 15—output displays loss only for the last observation because of loss statement outside the loop boundary.



Example 16:

```
data a;
p=100000;
int1=.09;
int2=.1
do i=1 to
5;
if i le 3
then
p+p*int1;
else
p+p*int2;
```

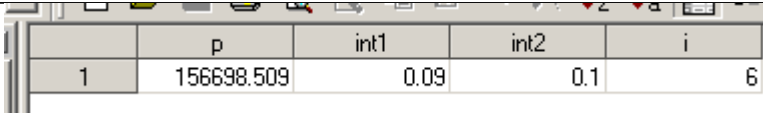
Example 17:

```
data a;
p=100000
;
int1=.09;
int2=.1
do i=1 to
5;
if i le 3
then
p+p*int1;
else p
```

Output

	p	int1	int2	i
1	109000	0.09	0.1	1
2	118810	0.09	0.1	2
3	129502.9	0.09	0.1	3
4	142453.19	0.09	0.1	4
5	156698.509	0.09	0.1	5

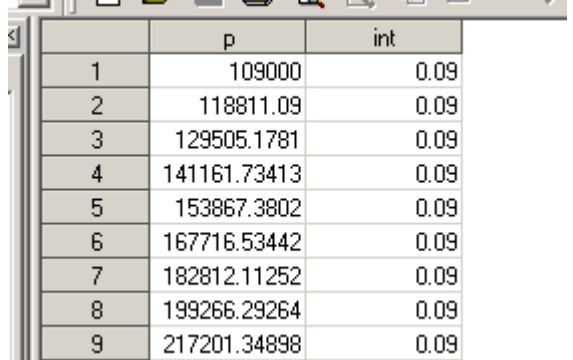
Ex: 16—output displays “P” calculated with interest rate = .09 for values of i < 3 and with interest rate = .1 for values of i >= 3.

output; end; run;	+p*int2; end; run;	 <p>Ex: 17—output calculates “P” with interest rate = .09 for values of i < 3 and with interest rate = .1 for values of i ≥ 3 but displays only the final value calculated of “p” when i = 5. Since there is no output statement within --do loop and gives result by implicit output at the end of loop. Note here, value of “i” has iterated to 6.</p>
-------------------------	--------------------------	--

```

data a;
p=100000;
int=.09;
do p=100000 to 200000;
p+p*int;
output;
end;
run;

```

Example 18:	Output
<pre> data a; P=100000; int = .09; do p=100000 to 200000; p+p*int; output; end; run; </pre>	 <p>This example helps in finding the number of years taken to get a final principal amount of 200000. The no. of observation implies no. of years taken to cross over the final amount if not exactly the same.</p>