```
/*Blanks in SAS:
In SAS, blanks refer to empty or missing values in a dataset. These blanks can occur when there is
no valid data available for a particular variable in an observation.

Types of Blanks in SAS:

Numeric Blanks: Numeric blanks occur when a numeric variable has a missing
 or empty value. In SAS, numeric blanks are represented by a period (.) or a missing value indicator.

Character Blanks: Character blanks occur when a character variable has a missing
or empty value. In SAS, character blanks are represented by an empty string ('') or a missing value indicator.

Dealing with Blanks in SAS:
To handle blanks in SAS programming, you can use various techniques and functions.
Here are some commonly used approaches:

Missing Value Functions:

MISSING(): The MISSING() function checks whether a variable has a missing value (blank) and
returns a boolean result.
Example: if missing(var) then put "Variable is missing.";
Coalescing Operator (??):

The coalescing operator (??) can be used to assign a default value to a variable if it is missing or blank.
Example: new_var = old_var ?? "Default Value";
IF-THEN Statements:

Use IF-THEN statements to conditionally check for blanks and perform specific actions based on the result.
Example:
sas
Copy code
if var = . then put "Variable is missing.";
else if var = '' then put "Variable is blank.";
FORMAT Statement:

You can use the FORMAT statement to assign a format to a variable that handles missing or blank values.
Example: format var $20.; (Assigns a character format of length 20 to the variable var)
Data Step Functions:

Functions like CATX(), CATS(), CATT(), COALESCE(), etc., can be used to concatenate or
manipulate variables while handling blanks appropriately.
Example: new_var = coalesce(var1, var2, "Default");
DROP or KEEP Statements:

Use the DROP or KEEP statements in a DATA step to exclude or retain variables with missing or blank values.
Example: drop var; (Excludes the variable var from the output dataset)
These are some simple techniques to handle blanks in SAS programming.
Introduce these concepts to your students gradually, starting with the basics,
and provide hands-on practice with examples and exercises to reinforce their understanding.

*/

/*Characte Blank*/
/*1.trailing Blank
  2.leading Blank
  3.Blanks in Between

  to deal with we use
  compress(remove all blank space) and compbl(remove extra blank space in characters)
*/

data sample;
name="Chetan    Rajaram        Salunke";
len=length(name);
compress=compress(name);
len2=length(compress);
compbl=compbl(name);
len3=length(compbl);
run;

data sample2;
values="aaAA123124020rundkjfnsad;n'oadhg09eq3%@#@^@";
len=length(values);
new_value=compress(values,'@1'); /*Remove character @ and 1 from original data*/
new_value2=compress(values,'a');/*Only remove small letter 'a' but not 'A'*/
new_value3=compress(values,'a','i');/* 'i' is modifier which consider neglect case sensitiveness*/
run;

/*String Concatenation function(key),cat(),catx() functions*/
```

```
data concate;
first_name="Chetan";
middle_name="Rajaram";
last_name="Salunke";
con=first_name||" "||middle_name||" "||last_name;
con2=cat(first_name,middle_name,last_name);
con3=catx(" ",first_name,middle_name,last_name);
run;



/*Scan(char,count,modifier) Function*/

data test;
input name$25.;
datalines;
Chetan_Rajaram_Salunke
Pradip_Shantaram_Fulpagare
Nikita_Sharad_Patil
Kanchan_Jayant_Mahajan
;

proc print data=test;
data test2;
set test;
first_name=scan(name,1,'_');
middle_name=scan(name,2,'_');
last_name=scan(name,3,'_');
proc print data=test2;
run;


/*Extracting the part of the string : substr() and  lowercase()*/
data card;
input card_number $1-50;
datalines;
123242343553t455
1232342342424624
2424248726487662
7987972646462462
9279737247246296
1642424249727272
;
proc print data=card;

data test_card;
set card;
sub=substr(card_number,2,10);
proc print data=test_card;



data test_card2;
set card;
sub=substr(card_number,2,10);
substr(card_number,2,10)='********';   /*value extracted replace by stars.*/
proc print data=test_card2;

data test3;
set test2;
new_name=upcase(name);
proc print data=test3;

data test4;
set test2;
new_name=lowcase(name);
proc print data=test4;

/*find and replace the words in the strings of data by tranwrd (transfer words)*/
 data ch;
 name="the boy is clever and boy don't like to involve in crowd";
 transfer_word_boy_by_girl=tranwrd(name,'boy','girl');
 proc print data=ch;



/*Correct Missinformation*/
data info;
input name$10. gender$10.;
datalines;
```

```
Mr.Chetan M
Mr.Nikita F
Ms.Pradip M
Mr.Kanchan F
;
proc print data=info;

data correct_info;
set info;
if upcase(gender)="M" then name2=tranwrd(name,'Ms',"Mr");
else if upcase(gender)="F" then name2=tranwrd(name,'Mr',"Ms");
else name2=name;
proc print data=correct_info;


/*Translate*/
data trial;
variable="situation not under control";
proc print data=trial;

data trial2;
set trial;
variable_new=translate(variable,"***","tua",);
proc print data=trial2;

/* index and find function*/
data prac;
set sashelp.baseball;
index=index(name,'Bill');
proc print data=prac(obs=50);

/*find( ) work similar to index() but it provide more features like modifiers */

/* index and find function*/
data prac;
set sashelp.baseball;
index=index(name,'Bill','i');
proc print data=prac(obs=50);



/*prxmatch () function*/
data pract;
input info$50.;
datalines;
chetan mo no: 9960923237
pradip mo no: 9373962417
kanchan mo no:9834149987
;
proc print data=pract;

/*How to extracts no.*/
data
```