1.A raw data file is listed below.

```
1---+----10---+----20---+---
son Frank 01/31/89
daughter June    12-25-87
brother Samuel 01/17/51
```

The following program is submitted using this file as input:

```
data work.family;
   infile 'file-specification';
   <insert INPUT statement here>
run;
```

Which INPUT statement correctly reads the values for the variable `Birthdate` as SAS date values?

```
a. input relation $ first_name $ birthdate date9.;
b. input relation $ first_name $ birthdate mmddyy8.;
c. input relation $ first_name $ birthdate : date9.;
d. input relation $ first_name $ birthdate : mmddyy8.;
```

Correct answer: d

An informat is used to translate the calendar date to a SAS date value. The date values are in the form of two-digit values for month-day-year, so the MMDDYY8. informat must be used. When using an informat with list input, the colon-format modifier is required to correctly associate the informat with the variable name.

You can learn about

- informats in **Reading Date and Time Values**
- the colon-format modifier in **Reading Free-Format Data**.

2.A raw data file is listed below.

```
1---+----10---+----20---+---
Jose,47,210
Sue,,108
```

The following SAS program is submitted using the raw data file above as input:

```
data employeestats;
   <insert INFILE statement here>
   input name $ age weight;
run;
```

The following output is desired:

name	age	weight
Jose	47	210
Sue	.	108

Which of the following INFILE statements completes the program and accesses the data correctly?

```
a. infile 'file-specification' pad;
b. infile 'file-specification' dsd;
```

```
c. infile 'file-specification' dlm=',';
d. infile 'file-specification' missover;
```

Correct answer: b

The PAD option specifies that SAS pad variable length records with blanks. The MISSOVER option prevents SAS from reading past the end of the line when reading free formatted data. The DLM= option specifies the comma as the delimiter; however, consecutive delimiters are treated as one by default. The DSD option correctly reads the data with commas as delimiters and two consecutive commas indicating a missing value like those in this raw data file.

You can learn about

- the PAD option in **Reading Raw Data in Fixed Fields**
- the MISSOVER option in **Creating Multiple Observations from a Single Record**
- the DLM= option and the DSD option in **Reading Free-Format Data**.

3.The following program is submitted:

```
data numrecords;
   infile cards dlm=',';
   input agent1 $ agent2 $ agent3 $;
cards;
jones,,brownjones,spencer,brown
;
run;
```

What is the value for the variable named `Agent2` in the second observation?

a. *brown*
b. *spencer*
c. '' (missing character value)
d. There is no value because only one observation is created.

Correct answer: d

The CARDS statement enables you to read instream data. Any number of consecutive commas are considered to be a single delimiter as a result of the DLM= option, and the length of each variable defaults to 8 bytes. Therefore, the values *jones*, *brownjon*, and *spencer* are assigned to `Agent1`, `Agent2`, and `Agent3`, respectively, for the first observation. The rest of the data on the record is not read by the INPUT statement and is not output to the data set.

You can learn about

- the CARDS statement in **Creating SAS Data Sets from Raw Data**
- the default length of variables in **Reading Free-Format Data**.

4.A raw data file is listed below.

```
1---+----10---+----20---+----30---+----40---+----50
TWOSTORY 1040 2      1SANDERS ROAD      $55,850
CONDO    2150 4    2.5JEANS AVENUE     $127,150
```

The following program is submitted using this file as input:

```
data work.houses;
   infile 'file-specification';
   <insert INPUT statement here>
run;
```

Which one of the following INPUT statements reads the raw data file correctly?

a.
```
      input @1  style $8.
      +1  sqfeet 4.
      +1  bedrooms 1.
      @20 baths 3.
          street 16.
      @40 price dollar8;
```

b.
```
      input @1  style $8
      +1  sqfeet 4.
      +1  bedrooms 1.
      @20 baths 3.
          street $16
      @40 price dollar8.;
```

c.
```
      input @1  style $8.
      +1  sqfeet 4.
      +1  bedrooms 1.
      @20 baths 3.
          street $16.
      @40 price dollar8.;
```

d.
```
      input @1  style $8.
      +1  sqfeet 4.
      +1  bedrooms 1.
      @20 baths 3
          street 16.
      @40 price dollar8.;
```

Correct answer: **c**

Formatted input requires periods as part of the informat name. The period is missing from the variables `Style` and `Street` in Answer *b*, the variable `Baths` in Answer *d*, and the variable `Price` in Answer *a* (which is also missing a dollar sign to read the variable `Street` as a character value).

You can learn about formatted input and informats in **Reading Raw Data in Fixed Fields**.

5. The following SAS program is submitted at the start of a new SAS session:

```
libname sasdata 'SAS-data-library';
data sasdata.sales;
   set sasdata.salesdata;
   profit=expenses-revenues;
run;
```

```
proc print data=sales;
run;
```

The SAS data set **Sasdata.Salesdata** has ten observations. Which one of the following explains why a report fails to generate?

a. The DATA step fails execution.
b. The SAS data set **Sales** does not exist.
c. The SAS data set **Sales** has no observations.
d. The PRINT procedure contains a syntax error.

Correct answer: b

The DATA step creates a permanent SAS data set, **Sasdata.Salesdata**. The PRINT procedure is printing a temporary SAS data set, **Sales**, that is stored in the **Work** library. At the beginning of the SAS session, **Work.Sales** does not exist.

You can learn about

- creating permanent data sets with the DATA step in **Creating SAS Data Sets from Raw Data**
- temporary data sets in **Basic Concepts**.

6. Which action assigns a reference named SALES to a permanent SAS data library?

a. Issuing the command:
```
libref SALES 'SAS-data-library'
```
b. Issuing the command:
```
libname SALES 'SAS-data-library'
```
c. Submitting the statement:
```
libref SALES 'SAS-data-library';
```
d. Submitting the statement:
```
libname SALES 'SAS-data-library';
```

Correct answer: d

The LIBNAME statement assigns a reference known as a libref to a permanent SAS data library. The LIBNAME command opens the LIBNAME window.

You can learn about the LIBNAME statement in **Referencing Files and Setting Options**.

7. The following SAS program is submitted:
```
data newstaff;
   set staff;
   <insert WHERE statement here>
run;
```
Which one of the following WHERE statements completes the program and selects only observations with a `Hire_date` of *February 23, 2000*?
```
a. where hire_date='23feb2000'd;
b. where hire_date='23feb2000';
c. where hire_date='02/23/2000'd;
```

d. `where hire_date='02/23/2000';`

Correct answer: a

A SAS date constant must take the form of one- or two-digit day, three-digit month, and two- or four-digit year, enclosed in quotation marks and followed by a **d** (*'ddmmmyy<yy>'*d).

You can learn about SAS date constants in **Creating SAS Data Sets from Raw Data**.

8. Which one of the following SAS date formats displays the SAS date value for January 16, 2002 in the form of 16/01/2002?

a. DATE10.
b. DDMMYY10.
c. WEEKDATE10.
d. DDMMYYYY10.

Correct answer: b

The requested output is in day-month-year order and is 10 bytes long, so DDMMYY10. is the correct format. Although WEEKDATE10. is a valid SAS format, it does not display the SAS date value as shown in the question above. DDMMYYYY10. is not a valid SAS date format, and the DATE*w*. format cannot accept a length of 10.

You can learn about

- the DDMMYY10. format in **Creating List Reports**
- the WEEKDATE10. format in **Reading Date and Time Values**.

9. Which one of the following displays the contents of an external file from within a SAS session?

a. the LIST procedure
b. the PRINT procedure
c. the FSLIST procedure
d. the VIEWTABLE window

Correct answer: c

The PRINT procedure and VIEWTABLE window display the values in SAS data sets. The FSLIST procedure displays the values in external files. There is no LIST procedure in SAS.

You can learn about

- the PRINT procedure in **Creating List Reports**

- the VIEWTABLE window in **Referencing Files and Setting Options**.

10. The SAS data set **Sashelp.Prdsale** contains the variables `Region` and `Salary` with 4 observations per `Region`. **Sashelp.Prdsale** is sorted primarily by `Region` and within `Region` by `Salary` in descending order.

The following program is submitted:

```
data one;
   set sashelp.prdsale;
   retain temp;
   by region descending salary;
      if first.region then
         do;
            temp=salary;
            output;
         end;
      if last.region then
         do;
            range=salary-temp;
            output;
         end;
run;
```

For each region, what is the number of observation(s) written to the output data set?

a. 0
b. 1
c. 2
d. 4

Correct answer: **c**

The expression `first.region` is true once for each region group. The expression `last.region` is true once for each region group. Therefore, each OUTPUT statement executes once for a total of 2 observations in the output data set.

You can learn about the FIRST.*variable* expression and the OUTPUT statement in **Reading SAS Data Sets**.

11. The following SAS program is submitted:

```
proc contents data=sasuser.houses;
run;
```

The exhibit below contains partial output produced by the CONTENTS procedure.

Data Set Name	SASUSER.HOUSES	**Observations**	15
Member Type	DATA	**Variables**	6
Engine	V9	**Indexes**	0
Created	Tuesday, April 22, 2003 03:09:25 PM	**Observation Length**	56
Last Modified	Tuesday, April 22, 2003 03:09:25 PM	**Deleted Observations**	0
Protection		**Compressed**	NO

Data Set Type		**Sorted**	NO
Label	Residential housing for sale		
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Which of the following describes the **Sasuser.Houses** data set?

a. The data set is sorted but not indexed.
b. The data set is both sorted and indexed.
c. The data set is not sorted but is indexed.
d. The data set is neither sorted nor indexed.

Correct answer: d

The exhibit above shows partial output from the CONTENTS procedure, In the top right-hand column of the output, you see that `Indexes` has a value of *0*, which indicates that no indexes exist for this data set. Also, `Sorted` has a value of *NO*, which indicates that the data is not sorted.

You can learn about the CONTENTS procedure in **Referencing Files and Setting Options**.

12. The following SAS program is submitted:

```
proc sort data=work.test;
   by fname descending salary;
run;
```

Which one of the following represents how the observations are sorted?

a. The data set **Work.Test** is stored in ascending order by both `Fname` and `Salary` values.
b. The data set **Work.Test** is stored in descending order by both `Fname` and `Salary` values.
c. The data set **Work.Test** is stored in descending order by `Fname` and ascending order by `Salary` values.
d. The data set **Work.Test** is stored in ascending order by `Fname` and in descending order by `Salary` values.

Correct answer: d

The DESCENDING keyword is placed before the variable name it modifies in the BY statement, so the correct description is in descending order by `Salary` value within ascending `Fname` values.

You can learn about the SORT procedure and the DESCENDING keyword in **Creating List Reports**.

13. The following SAS program is submitted:

```
data names;
```

```
   title='EDU';
   if title='EDU' then
      Division='Education';
   else if title='HR' then
      Division='Human Resources';
   else Division='Unknown';
run;
```

Which one of the following represents the value of the variable `Division` in the output data set?

a. Educatio
b. Education
c. Human Re
d. Human Resources

Correct answer: b

The length of the variable `Division` is set to 9 when the DATA step compiles. Since the value of the variable `Title` is *EDU*, the first IF condition is true; therefore, the value of the variable `Division` is *Education*.

You can learn about

- the length of a variable in **Understanding DATA Step Processing**
- IF-THEN statements in **Creating and Managing Variables**.

14. Which one of the following SAS programs creates a variable named `City` with a value of *Chicago*?

```
a.      data work.airports;
   AirportCode='ord';
   if AirportCode='ORD' City='Chicago';
run;
b.      data work.airports;
   AirportCode='ORD';
   if AirportCode='ORD' City='Chicago';
run;
c.      data work.airports;
   AirportCode='ORD';
   if AirportCode='ORD' then City='Chicago';
run;
d.      data work.airports;
   AirportCode='ORD';
   if AirportCode='ORD';
   then City='Chicago';
run;
```

Correct answer: c

The correct syntax for an IF-THEN statement is: **IF** *expression* **THEN** *statement*;
In this example, the variable `City` is assigned a value of *Chicago* only if the expression `AirportCode='ORD'` is true.

You can learn about IF-THEN statements in **Creating and Managing Variables**.

15. The following SAS program is submitted:

```
data work.building;
   code='DAL523';
   code='SANFRAN604';
   code='HOUS731';
   length code $ 20;
run;
```

Which one of the following is the length of the code variable?

a. 6
b. 7
c. 10
d. 20

Correct answer: **a**

The DATA step first goes through a compilation phase, then an execution phase. The length of a variable is set during the compilation phase and is based on the first time the variable is encountered. In this case, the variable code is set to the length of the text string *DAL523* which is 6 characters long. The next assignment statements are ignored during compilation. The LENGTH statement is also ignored since the length has already been established, but a note will be written to the log.

You can learn about

- the compilation phase of the DATA step in **Understanding DATA Step Processing**
- the LENGTH statement in **Creating and Managing Variables**.

16. Which of the following statements creates a numeric variable named IDnumber with a value of *4198*?

```
a. IDnumber=4198;
b. IDnumber='4198';
c. length IDnumber=8;
d. length IDnumber $ 8;
```

Correct answer: **a**

The first reference to the SAS variable in the DATA step sets the name, type, and length of the variable in the program data vector (PDV) and in the output SAS data set. The assignment statement `IDnumber=4198;` is the first reference and creates a numeric variable named IDnumber with a default storage length of 8 bytes.

You can learn about

- creating variables in the DATA step in **Understanding DATA Step Processing**

- numeric variables in **Basic Concepts**.

17. The following program is submitted:

```
data fltaten;
   input jobcode $ salary name $;
cards;
FLAT1 70000 Bob
FLAT2 60000 Joe
FLAT3 30000 Ann
;
run;
data desc;
   set fltaten;
   if salary>60000 then description='Over 60';
   else description='Under 60';
run;
```

What is value of the variable named `description` when the value for `salary` is 30000?

a. *Under 6*
b. *Under 60*
c. *Over 60*
d. `''` (missing character value)

Correct answer: **a**

The variable `description` is being created by the IF-THEN/ELSE statement during compilation. The first occurrence of the variable `description` is on the IF statement, and since it is assigned the value *Over 60*, the length of the variable is 7. Therefore, for the `salary` value of *30000*, `description` has the value of *Under 6* (the *0* is truncated.)

You can learn about

- the compilation phase of the DATA step in **Understanding DATA Step Processing**
- IF-THEN/ELSE statements in **Creating and Managing Variables**.

18. A raw data file is listed below.

```
1---+----10---+----20---+---
10
23
20
15
```

The following program is submitted:

```
data all_sales;
   infile 'file-specification';
   input receipts;
   <insert statement(s) here>
run;
```

Which statement(s) complete(s) the program and produce(s) a running total of the `Receipts` variable?

```
a. total+receipts;
b.      total 0;
 sum total;
c. total=total+receipts;
d. total=sum(total,receipts);
```

Correct answer: a

The SUM function and the assignment statement do not retain values across iterations of the DATA step. The sum statement `total+receipts;` initializes `total` to *0*, ignores missing values of `receipt`, retains the value of `total` from one iteration to the next, and adds the value of `receipts` to `total`.

You can learn about the sum statement in **Creating and Managing Variables**.

19.A raw data file is listed below.

```
1---+----10---+----20---+---
1901 2
1905 1
1910 6
1925 1
1941 1
```

The following SAS program is submitted and references the raw data file above:

```
data money;
   infile 'file-specification';
   input year quantity;
   total=total+quantity;
run;
```

What is the value of `total` when the data step finishes executing?

a. *0*
b. *1*
c. *11*
d. . (missing numeric value)

Correct answer: d

The variable `Total` is assigned a missing value during the compilation phase of the DATA step. When the first record is read in, SAS processes: `total=.+2;` which results in a missing value. Therefore the variable `Total` remains missing for all observations.

You can learn about

- the compilation phase of the DATA step in **Understanding DATA Step Processing**

- using missing values with arithmetic operators in **Creating SAS Data Sets from Raw Data**.

20. The following program is submitted:

```
data test;
   average=mean(6,4,.,2);
run;
```

What is the value of `average`?

a. *0*
b. *3*
c. *4*
d. . (missing numeric value)

Correct answer: **c**

The MEAN function adds all of the non-missing values and divides by the number of non-missing values. In this case, *6 + 4 + 2* divided by *3* is *4*.

You can learn about the MEAN function in **Transforming Data with SAS Functions**.

21. The following SAS program is submitted:

```
data work.AreaCodes;
   Phonenumber=3125551212;
   Code='('!!substr(Phonenumber,1,3)!!')';
run;
```

Which one of the following is the value of the variable `Code` in the output data set?

a. *(3)*
b. *(312)*
c. *3*
d. *312*

Correct answer: **a**

An automatic data conversion is performed whenever a numeric variable is used where SAS expects a character value. The numeric variable is written with the BEST12. format and the resulting character value is right-aligned when the conversion occurs. In this example, the value of `Phonenumber` is converted to character and right-aligned before the SUBSTR function is performed. Since there are only 10 digits in the value of `Phonenumber`, the right-aligned value begins with two blanks. Therefore the SUBSTR function picks up two blanks and a 3, and uses the BEST12. format to assign that value to `Code`. Then, the parentheses are concatenated before and after the two blanks and a 3.

You can learn about automatic data conversion and the SUBSTR function in **Transforming Data with SAS Functions**.

22. The following SAS program is submitted:

```
data work.inventory;
   products=7;
```

```
   do until (products gt 6);
      products+1;
   end;
run;
```

Which one of the following is the value of the variable `products` in the output data set?

a. *5*
b. *6*
c. *7*
d. *8*

Correct answer: **d**

A DO UNTIL loop always executes at least once because the condition is not evaluated until the bottom of the loop. In the SAS program above, the value of `Products` is incremented from *7* to *8* on the first iteration of the DO UNTIL loop, before the condition is checked. Therefore the value of `Products` is *8*.

You can learn about DO UNTIL loops in **Generating Data with DO Loops**.

23. The following program is submitted:

```
data work.test;
   set work.staff (keep=salary1 salary2 salary3);
   <insert ARRAY statement here>
run;
```

Which ARRAY statement completes the program and creates new variables?

```
a. array salary{3};
b. array new_salary{3};
c. array salary{3} salary1-salary3;
d. array new_salary{3} salary1-salary3;
```

Correct answer: **b**

Although each of the ARRAY statements listed above is a valid statement, only Answer B creates new variables named `new_salary1`, `new_salary2` and `new_salary3`. Answer C and Answer D both create an array that groups the existing data set variables `salary1`, `salary2`, and `salary3`. Since the array in Answer A is named `salary`, it also uses the existing data set variables.

You can learn about creating new variables in an ARRAY statement in **Processing Variables with Arrays**.

24. Which of the following permanently associates a format with a variable?

a. the FORMAT procedure
b. a FORMAT statement in a DATA step
c. an INPUT function with format modifiers
d. an INPUT statement with formatted style input

Correct answer: **b**

To permanently associate a format with a variable, you use the FORMAT statement in a DATA step. You can use the FORMAT procedure to create a user-defined format. You use the INPUT function to convert character data values to numeric values with an informat. You use the INPUT statement to read data into a data set with an informat.

You can learn about

- permanently assigning a format to a variable in **Creating and Managing Variables**
- the FORMAT statement in **Creating List Reports**
- the FORMAT procedure in **Creating and Applying User-Defined Formats**
- the INPUT function in **Transforming Data with SAS Functions**
- the INPUT statement in **Reading Raw Data in Fixed Fields**.

25. The following report is generated:

```
Style
of homes          n  Asking
Price
```

CONDO	4	$99,313
RANCH	4	$68,575
SPLIT	3	$77,983
TWOSTORY	4	$83,825

Which of the following steps created the report?

```
a.      proc freq data=sasuser.houses;
   tables style price /nocum;
   format price dollar10.;
   label style="Style of homes"
         price="Asking price";
run;
b.      proc print data=sasuser.houses;
   class style;
   var price;
   table style,n price*mean*f=dollar10.;
   label style="Style of homes"
         price="Asking price";
run;
c.      proc means data=sasuser.houses n mean;
   class style;
   var price;
   format price dollar10.;
   label style="Style of homes"
         price="Asking price";
run;
d.      proc report data=sasuser.houses nowd headline;
   column style n price;
   define style / group "Style of homes";
   define price / mean format=dollar8.
                  "Asking price";
run;
```

Correct answer: d

The FREQ procedure cannot create the average asking price. The CLASS statement and the VAR statement are not valid for use with the PRINT procedure. The MEANS procedure output would have both the N statistic and the N Obs statistic since a CLASS statement is used. The REPORT procedure produced the report.

You can learn about

- the FREQ procedure in **Producing Descriptive Statistics**
- the PRINT procedure in **Creating List Reports**
- the MEANS procedure in **Producing Descriptive Statistics**
- the REPORT procedure in **Creating Enhanced List and Summary Reports**.

26.A SAS report currently flows over two pages because it is too long to fit within the specified display dimension. Which one of the following actions would change the display dimension so that the report fits on one page?

a. Increase the value of the LINENO option.
b. Decrease the value of the PAGENO option.
c. Decrease the value of the LINESIZE option.
d. Increase the value of the PAGESIZE option.

Correct answer: d

The PAGESIZE= SAS system option controls the number of lines that compose a page of SAS procedure output. By increasing the number of lines available per page, the report might fit on one page.

You can learn about the PAGESIZE= option in **Referencing Files and Setting Options**.

27. Which one of the following SAS REPORT procedure options controls how column headings are displayed over multiple lines?

a. SPACE=
b. SPLIT=
c. LABEL=
d. BREAK=

Correct answer: b

The SPLIT= option specifies how to split column headings. The SPACE=, LABEL= and BREAK= options are not valid options in PROC REPORT.

You can learn about the SPLIT= option for the REPORT procedure in **Creating Enhanced List and Summary Reports**.

28. The following SAS program is submitted:

```
ods html file='newfile.html';
proc print data=sasuser.houses;
run;
proc means data=sasuser.houses;
run;
proc freq data=sasuser.shoes;
run;
ods html close;
proc print data=sasuser.shoes;
run;
```

How many HTML files are created?

a. 1
b. 2
c. 3
d. 4

Correct answer: a

By default, one HTML file is created for each FILE= option or BODY= option in the ODS HTML statement. The ODS HTML CLOSE statement closes the open HTML file and ends the output capture. The **Newfile.html** file contains the output from the PRINT, MEANS, and FREQ procedures.

You can learn about the ODS HTML statement in **Producing HTML Output**.

29.A frequency report of the variable `Jobcode` in the **Work.Actors** data set is listed below.

Jobcode	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Actor I	2	33.33	2	33.33
Actor II	2	33.33	4	66.67
Actor III	2	33.33	6	100.00

Frequency Missing = 1

The following SAS program is submitted:

```
data work.joblevels;
   set work.actors;
   if jobcode in ('Actor I', 'Actor II') then
      joblevel='Beginner';
   if jobcode='Actor III' then
      joblevel='Advanced';
   else joblevel='Unknown';
run;
```

Which of the following represents the possible values for the variable `joblevel` in the **Work.Joblevels** data set?

a. *Advanced* and *Unknown* only

b. *Beginner* and *Advanced* only
c. *Beginner*, *Advanced*, and *Unknown*
d. '' (missing character value)

Correct answer: a

The DATA step will continue to process those observations that satisfy the condition in the first IF statement Although `Joblevel` might be set to *Beginner* for one or more observations, the condition on the second IF statement will evaluate as false, and the ELSE statement will execute and overwrite the value of `Joblevel` as *Unknown*.

You can learn about

- the IF statement in **Creating SAS Data Sets from Raw Data**
- the ELSE statement in **Creating and Managing Variables**.

30. The descriptor and data portions of the **Work.Salaries** data set are shown below.

Variable	Type	Len	Pos
name	Char	8	0
salary	Char	8	16
status	Char	8	8

name	status	salary
Liz	S	15,600
Herman	S	26,700
Marty	S	35,000

The following SAS program is submitted:

```
proc print data=work.salaries;
   where salary<20000;
run;
```

What is displayed in the SAS log after the program is executed?
a. A NOTE indicating that 1 observation is read.
b. A NOTE indicating that 0 observations were read.
c. A WARNING indicating that character values have been converted to numeric values.
d. An ERROR indicating that the WHERE clause operator requires compatible variables.

Correct answer: d

`Salary` is defined as a character variable. Therefore, the value in the WHERE statement must be the character value *20,000* enclosed in quotation marks.

You can learn about the WHERE statement in **Creating List Reports**.

31. Which of the following statements is true when SAS encounters a syntax error in a DATA step?

a. The SAS log contains an explanation of the error.
b. The DATA step continues to execute and the resulting data set is complete.
c. The DATA step stops executing at the point of the error and the resulting data set contains observations up to that point.
d. A note appears in the SAS log indicating that the incorrect statement was saved to a SAS data set for further examination.

Correct answer: a

SAS scans the DATA step for syntax errors during the compilation phase. If there are syntax errors, those errors get written to the log. Most syntax errors prevent further processing of the DATA step.

You can learn about how SAS handles syntax errors in the DATA step in **Understanding DATA Step Processing**.

32. Which TITLE statement would display JANE'S DOG as the text of the title?

```
a. title "JANE"S DOG";
b. title 'JANE"S DOG';
c. title "JANE'S DOG";
d. title 'JANE' ' 'S DOG';
```

Correct answer: c

The title in a TITLE statement must be enclosed in a pair of matched quotation marks. Unbalanced quotation marks can cause problems for SAS. To hide an unmatched single quotation mark, surround the title text with matched double quotation marks.

You can learn about

- the TITLE statement in **Creating List Reports**
- unbalanced quotation marks in **Editing and Debugging SAS Programs**.

33. The following SAS program is submitted:

```
data test;
   input animal1 $ animal2 $
         mlgrams1  mlgrams2;
cards;
hummingbird ostrich 54000.39 90800000.87
;
run;
```

Which one of the following represents the values of each variable in the output data set?

a.
```
animal1 animal2   mlgrams1   mlgrams2
hummingb          ostrich 54000.39    90800000
```

b.
```
animal1 animal2   mlgrams1   mlgrams2
hummingb          ostrich 54000.39    90800000.87
```

c.
```
  animal1 animal2    mlgrams1 mlgrams2
  hummingbird        ostrich  54000.39   90800000
```

d.
```
  animal1 animal2    mlgrams1 mlgrams2
```

hummingbird ostrich 54000.39 90800000.87

Correct answer: b

The CARDS statement is an alias for the DATALINES statement. In the INPUT statement, you must specify a dollar sign ($) after the variable name in order to define a character variable. If you do not specify otherwise, the default storage length for a variable is 8. In the example above, the character value *hummingbird* is truncated to *hummingb*.

You can learn about

- the DATALINES statement in **Creating SAS Data Sets from Raw Data**
- the INPUT statement in **Reading Free-Format Data**
- the default storage length for variables in **Basic Concepts**.

34. The SAS data sets **Work.Employee** and **Work.Salary** are shown below.

Work.Employee

fname	age
Bruce	30
Dan	40

Work.Salary

fname	salary
Bruce	25000
Bruce	35000
Dan	25000

The following merged SAS data set is generated:

Work.Empdata

fname	age	totsal
Bruce	30	60000
Dan	40	25000

Which one of the following SAS programs created the merged data set?

a.
```
      data work.empdata;
   merge work.employee
         work.salary;
   by fname;
   if first.fname then totsal=0;
   totsal+salary;
   if last.fname then output;
run;
```
b.
```
      data work.empdata(drop=salary);
   merge work.employee
         work.salary;
   by fname;
```

```
   if first.fname then totsal=0;
   totsal+salary;
   if last.fname then output;
run;
```

c.
```
   data work.empdata;
   merge work.employee
         work.salary(drop=salary);
   by fname;
   if first.fname then total=0;
   totsal+salary;
   if last.fname then output;
run;
```

d.
```
   data work.empdata;
   merge work.employee
         work.salary;
   by fname;
   if first.fname then total+salary;
run;
```

Correct answer: b

The MERGE and BY statements allow you to match-merge two or more SAS data sets. The BY statement creates two temporary variables, `First.Fname` and `Last.Fname` for BY group processing. The SUM statement is used to add salary for each BY group. The variable on the left side of the plus sign is the variable that is retained and has an initial value of 0. The expression on the right side of the plus sign is added to the variable on the left side of the plus sign to create a grand total. The accumulating variable, `totsal`, is reset back to *0* when you encounter a new BY group value (`First.Fname` is true). To output just the totals for each BY group, use the explicit OUTPUT statement when you reach the last occurrence of each `Fname` value.

You can learn about the MERGE statement, the BY statement and match-merging in **Combining SAS Data Sets**.

35. The contents of the SAS data set **Sasdata.Group** are listed below.

name	age
Janice	10
Henri	11
Michele	11
Susan	12

The following SAS program is submitted using the **Sasdata.Group** data set as input:

```
libname sasdata 'SAS-data-library';
data group;
   set sasdata.group;
   file 'file-specification';
   put name $15. @5 age 2.;
run;
```

Which one of the following describes the output created?

a. a raw data file only

b. a SAS data set named **Group** only
c. both a SAS data set named **Group** and a raw data file
d. The program fails execution due to errors.

Correct answer: **c**

The DATA step creates a temporary data set named **Group** and reads data from the permanent SAS data set named **Sasdata.Group** into it. The FILE statement creates a raw data file by writing out values for the variables `Name` and `Age` to the file that is specified within the quotation marks.

You can learn about

- temporary data sets in **Basic Concepts**
- the FILE statement in **Creating SAS Data Sets from Raw Data**.

36. The SAS data set **Employee_info** is listed below.

employee	bonus
2542	100.00
3612	133.15
2198	234.34
2198	111.12

The following SAS program is submitted:

```
proc sort data=employee_info;
   <insert BY statement here>
run;
```

Which one of the following BY statements completes the program and sorts the data in sequential order by descending `bonus` values within ascending `employee` values?

```
a. by descending bonus employee;
b. by employee bonus descending;
c. by employee descending bonus;
d. by descending employee bonus;
```

Correct answer: **c**

You use the keyword DESCENDING in a BY statement to specify that data will be sorted by values in descending order. The DESCENDING keyword applies to the variable that is listed immediately after it. To sort on values of `bonus` within sorted values of `employee`, you list `employee` first in the BY statement.

You can learn about the DESCENDING keyword and the BY statement in **Creating List Reports**.

37. The following SAS program is submitted:

```
data work.accounting;
   length jobcode $ 12;
   set work.department;
```

```
run;
```

The **Work.Department** SAS data set contains a character variable named `jobcode` with a length of 5. Which of the following is the length of the variable `jobcode` in the output data set?

a. 5
b. 8
c. 12
d. The value cannot be determined because the program fails to execute due to syntax errors.

Correct answer: c

The LENGTH statement enables you to specify the length of a character variable. Since the LENGTH statement appears before the SET statement, SAS will set the length of `jobcode` to 12.

You can learn about the LENGTH statement in **Creating and Managing Variables**.

38. Assume the SAS data set **Sasuser.Houses** has four numeric variables.

The following SAS program is submitted:

```
proc means data=sasuser.houses mean;
   <insert statement(s) here>
run;
```

The following report is produced:

style	N Obs	Variable	Mean
CONDO	4	bedrooms	
baths	2.7500000		
2.1250000			
RANCH	4	bedrooms	
baths	2.2500000		
2.0000000			
SPLIT	3	bedrooms	
baths	2.6666667		
1.8333333			
TWOSTORY	4	bedrooms	
baths	3.0000000		
1.8750000			

Which of the following statement(s) create(s) this report?

```
a. class style;
b. var bedrooms baths;
c.      class style;
var bedrooms baths;
d.      var style;
class bedrooms baths;
```

Correct answer: c

The CLASS statement specifies the category variable(s) for group processing. The VAR statement specifies the numeric variable(s) for which to calculate statistics.

You can learn about the CLASS statement and the VAR statement in **Producing Descriptive Statistics**.

39. An HTML file contains a SAS report. Which ODS statement option is used to specify the name of the HTML file?

a. OUT=
b. FILE=
c. HTML=
d. HTMLFILE=

Correct answer: b

The FILE= option identifies the file that contains the HTML output. The FILE= option is an alias for the BODY= option in the ODS HTML statement.

You can learn about the FILE= option in the ODS HTML statement in **Producing HTML Output**.

40. The following SAS program is submitted:

```
data work.test;
   set sasuser.class;
   array t{3} <insert text here> (5, 10, 15);
run;
```

Which one of the following completes the ARRAY statement and creates data elements that are not included in the SAS data set **Work.Test**?

a. _DROP_
b. _TEMP_
c. _TEMPORARY_
d. No extra text is needed.

Correct answer: c

TEMPORARY is a keyword used in the ARRAY statement to create temporary data elements. By default, the ARRAY statement creates new data set variables or references existing variables for use by the array.

You can learn about the _TEMPORARY_ keyword and the ARRAY statement in **Processing Variables with Arrays**.

41.A raw data file is listed below.

```
1---+----10---+----20---+---
01/05/1989    Frank     11
12/25/1987    June      13
01/05/1991    Sally      9
```

The following SAS program is submitted using the raw data file as input:

```
data work.family;
   infile 'file-specification';
   input @1 date_of_birth mmddyy10.
         @15 first_name $5.
         @25 age 3;
run;

proc print data=work.family noobs;
run;
```

Which one of the following is the result?

a. The program executes, but the age values are missing in the output.
b. The program executes, but the date values are missing in the output.
c. The program fails to execute because the age informat is coded incorrectly.
d. The program fails to execute because the date informat is coded incorrectly.

Correct answer: a

Values for the variable age are missing in the output because the informat for age is coded incorrectly. Since age is standard numeric input, it should use the *w.d* informat to specify a field width of 3 in the INPUT statement.

You can learn about the *w.d* informat in **Reading Raw Data in Fixed Fields**.

42. Assume that SAS data sets **Sasdata.Products** and **Sasdata.Sales** both contain the Prod_ID variable. Which of the following SAS DATA steps returns only exceptions or non matches?

a.
```
      libname sasdata 'SAS-data-library';
data all;
   merge sasdata.products
         sasdata.sales;
   by prod_id;
   if ins=1 or inp=1;
run;
```

b.
```
      libname sasdata 'SAS-data-library';
data all;
   merge sasdata.products(in=inp)
         sasdata.sales(in=ins);
   by prod_id;
   if ins=1 and inp=1;
run;
```

c.
```
      libname sasdata 'SAS-data-library';
data all;
   merge sasdata.products(in=inp)
         sasdata.sales(in=ins);
   by prod_id;
   if ins=0 and inp=0;
run;
```

d.
```
      libname sasdata 'SAS-data-library';
data all;
   merge sasdata.products(in=inp)
```

```
        sasdata.sales(in=ins);
   by prod_id;
   if ins=0 or inp=0;
run;
```

Correct answer: d

By default, DATA step match-merges combine all observations in all input data sets. To include only unmatched observations from your output data set, you can use the IN= data set option to create and name a temporary variable that indicates whether the data set contributed to the current observations. If the value of the IN= variable is *0*, the data set did not contribute to the current observation; if the value is *1*, the data set did contribute to the current observation. You can use a subsetting IF statement to only include those observations that have a value of *0* for the IN= variable of either the **Sasdata.Products** data set or the **Sasdata.Sales** data set. Since an unmatched observation might come from either input data set, you do not need to specify that the IN= variables for both **Sasdata.Products** and **Sasdata.Sales** have values of *0*.

You can learn about the IN= data set option in **Combining SAS Data Sets**.

43. The following SAS program is submitted:

```
libname sasdata 'SAS-data-library';
libname labdata 'SAS-data-library';
data labdata.boston
     labdata.dallas(drop=city dest equipment);
   set sasdata.cities(keep=orig dest city
                      price equipment);
   if dest='BOS' then output labdata.boston;
   else if dest='DFW' then output labdata.dallas;
run;
```

Which variables are output to both data sets?

a. `price` and `orig` only
b. `city` and `equipment` only
c. `city`, `price`, and `equipment` only
d. `city`, `price`, `orig`, and `equipment`

Correct answer: a

In the program above, the KEEP= option specifies that 5 variables (`orig`, `dest`, `city`, `price`, and `equipment`) are read from the input data set. All of these variables are output to **Labdata.Boston**. In the **Labdata.Dallas** output data set, the DROP= option specifies that `city`, `dest`, and `equipment` are excluded from the data set so that only `orig` and `price` are included.

You can learn about the KEEP= option in **Creating and Managing Variables**.

44. The following SAS program is submitted:

```
proc contents data=sasuser._all_ nods;
run;
```

Which one of the following is produced as output?
a. the list of all data set names in the **Sasuser** library only
b. the descriptor portion of the data set named **Sasuser._All_**
c. the descriptor portion of every data set in the **Sasuser** library only
d. the list of data set named in the **Sasuser** library plus the descriptor portion of every data set in the **Sasuser** library.

Correct answer: a

You can use the CONTENTS procedure to create SAS output that describes the contents of a library. _ALL_ requests a listing of all files in the library, and NODS suppresses the printing of detailed information about each file in the output.

You can learn about the CONTENTS procedure in **Referencing Files and Setting Options**.

45. The following SAS program is submitted:

```
data work.test;
   length city $20;
   city='Paris';
   city2=trim(city);
run;
```

Which one of the following is the length of the `city2` variable?
a. 5
b. 6
c. 8
d. 20

Correct answer: d

The LENGTH statement specifies that the variable `city` has a length of 20 characters. The TRIM function in the assignment statement for `city2` removes trailing blanks from the value. However, the length of the `city2` variable is set to 20 because `city` has a length of 20, and SAS pads the trimmed value with extra blanks.

You can learn about

- the LENGTH statement in **Reading Free-Format Data**
- the TRIM function in **Transforming Data with SAS Functions**.

46. A raw data record is listed below.

```
1---+----10---+----20---+---
$23,456 750
```

The following SAS program is submitted:

```
data bonus;
```

```
   infile 'file-specification';
   input salary $ 1-7 raise 9-11;
   <insert statement here>
run;
```

Which one of the following statements completes the program and adds the values of `salary` and `raise` to calculate the expected values of the `newsalary` variable?

a. `newsalary=salary + raise;`
b. `newsalary=put(salary,comma7.) + raise;`
c. `newsalary=input(salary,comma7.) + raise;`
d. `newsalary=put(salary,comma7.) + put(raise,3.);`

Correct answer: **c**

You must convert the value of `salary` from character to numeric in order to perform an arithmetic function on it. You use the INPUT function to convert a character value to a numeric value.

You can learn about the INPUT function in **Transforming Data with SAS Functions**.

47. The following SAS program is submitted:

```
data work.travel;
   do i=1 to 6 by 2;
      trip + i;
   end;
run;
```

Which one of the following is the value of the variable `trip` in the output data set?

a. *2*
b. *3*
c. *9*
d. *10*

Correct answer: **c**

The sum variable in a sum statement is automatically set to *0* before the first observation is read from the data set. The sum variable in the statement above is increased by the value of `i` on each iteration of the DO loop; the value of i is set to *1* on the first iteration of the DO loop and increases by *2* on each additional iteration until it is greater than *6*. Therefore, the value of `Trip` is equal to *0 + 1 + 3 + 5*, which is *9*.

You can learn about

- the sum statement in **Creating and Managing Variables**
- DO loops in **Generating Data with DO Loops**.

48. The following SAS program is submitted:

```
proc report data=survey nowd;
   column age choice1;
   <insert DEFINE statement here>
   define choice1/display;
```

```
run;
```

Which one of the following DEFINE statements completes the program and displays values of the variable Age in ascending order?

```
a. define age/sort;
b. define age/order;
c. define age/sort by age;
d. define age/order by age;
```

Correct answer: b

You use a DEFINE statement to describe how to order and use variables in your report. To specify Age as an order variable, you use the ORDER usage option in the DEFINE statement. An order variable orders the detail rows in a report according to their formatted values. By default, the order is ascending.

You can learn about the DEFINE statement and order variables in **Creating Enhanced List and Summary Reports**.

49. Which one of the following statements is true when SAS encounters a data error?

a. The execution phase is stopped, and a system abend occurs.
b. A missing value is assigned to the appropriate variable, and execution continues.
c. The execution phase is stopped, and a SAS data set is created with zero observations.
d. A missing value is assigned to the appropriate variable, and execution stops at that point.

Correct answer: b

Unlike syntax errors, invalid data errors do not cause SAS to stop processing a program. SAS handles invalid data errors by assigning a missing value to the appropriate variable and writing a message to the SAS log.

You can learn about how SAS handles invalid data errors in **Creating SAS Data Sets from Raw Data**.

50. The following SAS program is submitted:

```
data test;
   input country $8. date mmddyy10.;
cards;
Germany 12/31/2000
France 01/32/2001
;
run;
```

Which one of the following is the value of the variable _ERROR_ when the variable _N_ has a value of *2*?

a. 0
b. 1
c. true
d. false

Correct answer: **b**

_N_ and _ERROR_ are automatic variables that can be used for DATA step processing but that are not written to the output data set. _N_ counts the number of times that the DATA step begins to execute, and _ERROR_ signals the occurrence of an error that is caused by the data during execution. A value of *0* indicates no error while a value of *1* indicates one or more errors. In the program above, the value for date in the second observation is invalid.

You can learn about the automatic variables _N_ and _ERROR_ in **Understanding DATA Step Processing**.