MAXIMIZING SAS® TRAINING
Joe Dingwerth, Mutual of Omaha Insurance

I) Course Materials And Retention

A) 'Skill-based' vs. 'Reference-based'
    Information Transfer

Many in-house SAS course handouts rely on excerpts from the manual to convey information. While manuals are invaluable, handouts may be structured to provide answers to WHAT, HOW, WHEN, WHY, and WHERE questions about the topic. This changes the focus from "reference-based" delivery to "skill-based", thus facilitating retention. Which of the following styles best exemplifies your training handouts?

1A) Example of 'Reference-based' handout style:

PROC FORMAT:
You can define formats to your own specifications with the format procedure. PROC FORMAT produces two kinds of formats:

Value-labeling formats associate labels with values. The VALUE statement generates these formats. For example, you might store SEX as a numeric code, 1 or 2, but print it as a character value using the following statement:

    VALUE SEX 1='MALE' 2='FEMALE';

2A) Example of 'Skill-based' handout style:

PROC FORMAT - a closer look at the VALUE statement:

What does it do?
PROC FORMAT allows you to create your own custom formats. One of the statements within PROC FORMAT is the VALUE statement. VALUE lets you create either numeric or character formats. Let's first look at character formats.

Why use it?
In many data files, some of the information is encoded (employee numbers, product codes, department codes, etc.) Without the use of PROC FORMAT, the reports containing these coded fields are difficult to understand. With PROC FORMAT, however, you can format these coded fields into whatever character translation you desire.

Where does it go?
PROC FORMAT is an unusual PROC because it can go before the DATA step. In fact, that is a traditional place to put it - at the beginning of the program, because PROC FORMAT only creates your custom formats; it does not tell SAS to use them. Therefore, it poses no harm to the DATA step and can go before it.

When do I use it?
By using the FORMAT statement and referring to both the variable and your new format name, SAS will use your format in the same way that it uses any other format (DOLLAR, COMMA, etc.).

How do I write it?
PROC FORMAT has two statements: VALUE and PICTURE. VALUE can work with either character or numeric fields, while PICTURE is limited to numeric fields. Since we're just looking at character formats right now, let's examine the VALUE statement more closely.
VALUE statement:

Imagine you have a character field called REPNUM that contains salesrep ID numbers (3319, 1178). You want the reps' real names to print out instead of their ID numbers. Since the field REPNUM is character, you create a 'character' format as seen below:

OPTIONS CENTER;

PROC FORMAT;
  VALUE REPNUM;
  3319 = 'Timothy Bottom';
  1178 = 'Joan River';
DATA TEM:
  SET SASDATA.SALES;
INPUT @DOL POLNUM $07.
  @O20 ADDRESS $20.
  @O50 QUANTITY 6.2
  @O70 PRICE 5.2
  @134 REPNUM $04. ;
TOTAL = QUANTITY * PRICE;
PROC PRINT DATA=TEMP SPUT = 'j'
  NOOBS:
  VAR POLNUM REPNUM QUANTITY PRICE TOTAL;
  FORMAT TOTAL DOLLAR16.2
  LABEL POLNUM = 'POL/NO.'
  REPNUM = 'REP/NAME'
  QUANTITY = 'QTY.'
  PRICE = '/PRICE'
  TOTAL = 'TOTAL/COST';

(Approximation of LISTING):

<table>
<thead>
<tr>
<th>POL. NO.</th>
<th>REP. NAME</th>
<th>QTY.</th>
<th>PRICE</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>A6672</td>
<td>Timothy Bottom</td>
<td>100</td>
<td>.50</td>
<td>$50.00</td>
</tr>
<tr>
<td>BC132</td>
<td>Timothy Bottom</td>
<td>250</td>
<td>.75</td>
<td>$187.50</td>
</tr>
<tr>
<td>BC2079</td>
<td>Joan River</td>
<td>853</td>
<td>.65</td>
<td>$554.45</td>
</tr>
</tbody>
</table>

Recap:

1) PROC FORMAT goes before the DATA step.

2) the format name we created refers to a character field, which is why we put a dollar sign at the start of the format name.

3) to define the format, you list the current content of the field in quotes, type in an equal sign, and then enter your translation in quotes.

4) to activate the format, you use the FORMAT statement. In it, you list the variable name (REPNUM) and then your custom format name ($REP), followed immediately by a period to signal to SAS that it is a format.

5) how much easier it is to understand REPNUM (under the header REP. NAME) in its new format than the unformatted POLNUM (under the header POL. NO.).

B) Handout Packaging Basics

Often, we trainers find ourselves 'under the gun' to put a handout together, and we forget to include some essential tools for the student. Listed below are some key components to any effective handout:

1) Page numbering on ALL pages, including Appendices (which should have an alternate numbering scheme i.e. Roman Numerals).

2) Table of contents at the beginning of the handout, with section headings and page numbers.

3) Easy-to-read printer fonts. As you can see, getting everything you want on one page isn't as important as being able to read it!

4) Accurate, up-to-date information. Outdated information, even in a few places, tends to discredit the entire handout. After class, students may simply not use the handout rather than try to remember which information is accurate and which is not.

C) Reference Cards/Cheat Sheets/Job Aids

Cheat Sheets

Problems:

The mini-reference card has gained enormous popularity among computer users because of its handiness in finding quick answers to common questions. This simplicity, however, also can cause a problem: when the user needs an explanation not found on the 'cheat sheet', he/she calls the local Help Desk or MIS area.
A Solution:
One method of increasing the effectiveness of 'cheat sheets' is to incorporate ongoing references to more complete documentation, such as manuals or handouts. Include section or page numbers which point to broader descriptions or in-depth treatments of the topics included on the reference card.

Customized Cheat Sheets:
A creative way of distributing cheat sheets is to make them part of the 'graduation' ritual at the end of SAS class(es). Have the graduating student run an interactive SAS program that prompts them to enter their name and presents a checklist of topics and functions which were covered during the class. The student indicates the functions they are likely to use and the SAS program prints out a personalized cheat sheet, along with appropriate manual references and support telephone numbers.

POST-IT® Note Job Aids
I have heard of some companies that have printed mini-job aids on POST-IT® notes to encourage users to keep them as close to the terminal as possible.

D) Screen captures

More and more companies are using PC's with mainframe emulation instead of the traditional mainframe terminals, because of the flexibility it provides the user. This new technology also provides an invaluable tool for the technical trainer: the ability to 'capture' mainframe screens as ASCII files, which can then be incorporated into your course handouts.

These screen captures can be in:
- course exercises (Source Code, Listings)
- procedural training (showing SAS/AF® screens and what to type in to run a specific program)
- updates (SAS enhancements, in-house changes) can be announced in a memorandum format by including the captured screens in the memo. This reduces or eliminates the need for live discussion or Enhancement Seminars.

In addition to the above uses to you as a SAS trainer, screen captures provide reinforcement to the student that he/she is in the right place, doing the right thing. This reduces stress, and also provides a permanent takeback element in the event that the student wants to review course material at a later date.

On the adjacent column are some examples of screen captures and their effectiveness:

EXERCISE: Complete the program below by replacing the lower-case 'x's with appropriate code.

```
DATA SET WORK.HOUSEHOLDALLOCS (LABEL = "Household Allocations");
INPUT banks 1-3 / LABEL = "Bank Numbers";
INPUT index 4-6 / LABEL = "Index Numbers";
INPUT locality 7-10 / LABEL = "Locality Numbers";
INPUT type 11-12 / LABEL = "Type Numbers";
INPUT size 13 / LABEL = "Household Size";
RUN;
```
E) 'Smile Sheets' VS. Course Evaluations: the VCP™ Evaluation Style

How many times have we had to fill out an evaluation sheet where the primary focus of the questions had little to do with the effectiveness of the course? These 'smile sheets' still have their place as a means of evaluating the instructor, but they do little to evaluate the class, specifically, to measure the level of student comprehension. Elliot Masie's VCP™ (Vocabulary, Concept and Procedures) Listing provides an effective alternative to the more "traditional" evaluation tools.

Rather than ask the participants if they "liked the class" or if they thought the "instructor was effective", the VCP focuses on specific skills and software functions. The forms provide the learner with a listing of key vocabulary words, concepts and procedures which were presented in class. Towards the end of class the instructor asks each learner to read through the list and place a check mark under the column heading which best describes their level of understanding (or level of confusion) about each individual topic. This serves two functions: a self-inventory and a form of specific feedback for the instructor and training department.

The VCP method of evaluation moves the focus from the performance of the instructor and the mood of the classroom to specific skills and knowledge. Groups using this method have reported that students enjoy the ability to classify their level of confusion and it is an ideal vehicle for structuring follow-up.

Some tips for effective use of the VCP as an end-of-class evaluation tool are listed below:

1) Keep the list to one page if possible. The learner will feel exhausted if confronted with a five page VCP listing.

2) Develop column headings that make sense for your setting. Some organizations have a continuum from Fully Understand to Fully Confused. Another column heading range could be from Will Definitely Use to Have No Need To Use.

An example of Masie's VCP using a "SAS Fundamentals Course" evaluation instrument as an illustration can be found below:

<table>
<thead>
<tr>
<th>Vocabulary:</th>
<th>Fully Understand</th>
<th>I am slightly confused</th>
<th>I do not understand</th>
<th>Was not addressed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steps</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Statements</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Observations</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variables</td>
<td>x</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concepts:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Temporary SAS Data Sets</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Permanent SAS Data Sets</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Informats and Formats</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAS Functions, Dates</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Program Data Vector</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedures:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How to read in data from an external file (INPUT)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How to set conditional branching (IF THEN ELSE)</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How to write an assignment statement</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>How to perform DATA Step processing to write a custom report</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
II) Six ‘Road-tested’
Retention Techniques

1) Letter to Self

Upon the completion of class, hand each student a
blank sheet of paper and a routing envelope. Ask
them to put their routing address on the envelope.
Then, instruct them to write themselves a note. The
content of the letter is totally up to the individual, but
suggest that it deal with something learned in class. If
they look confused, give them an example such as:

“When I get back to my area, I’m going to practice
what I’ve learned immediately. I’m also going to go
over the exercises and the course handouts from time
to time to keep myself from forgetting. Finally, I’m
going to try to put some PROC FORMATS in my
programs to make them easier to understand.”

The ‘letter to self’ is a form of self-contract. Have the
students place their ‘note’ inside the envelope and
leave them with you. In approximately 6-8 weeks,
send them back to the students. The effect of
receiving a note from yourself is quite unsettling,
especially if you haven’t done what you told yourself
you would do. You can’t fool yourself!

2) Exercises / ‘Quizzes’ sent to students
4-6 weeks after class

From time to time, put together a short ‘quiz’ on a
SAS topic covered in class. Preface the ‘quiz’ with a
short note, conveying to the reader that you hope
he/she hasn’t forgotten everything from class. And to
prove it, take a few minutes to solve the following
problem. It is important to have a “bridging memo”
that explains the quiz’s purpose.

NOTE: Don’t make this too hard, and be sure the
solution involves applying methods which were
covered in class. The goal is to challenge the reader
without intimidating them. Print the answer on the
opposite side of the page. You may even want to use
an actual exercise from the class, with slight
modifications. Take pains to make the correspondence appealing. Printing it on bright paper
may make it more difficult to forget or lose in the
desktop shuffle.

3) Board games

Adults learn best when they’re not aware that they’re
learning because they put up no emotional or
psychological barriers. One ‘devious’ approach to
teaching SAS, therefore, is via a board game. Don’t
recreate the wheel - use an existing board game like
“Clue” or “Monopoly.”

Let’s use “Monopoly” as an example. The teacher
hands out a SAS LISTING printout. The goal of the
game is to build a SAS program that will generate the
LISTING by ‘buying’ up the right code.

a) First, tape over all of the Monopoly squares and
replace them with assorted SAS statements.
However, keep the dollar value of the square as it
appears in the real game. These SAS
statements can be any statement you can think of.
The code for the solution should be intermixed within them, and there should be
some duplication to make the game go faster.
When the player lands on a square, he/she can
buy it if the statement will help in the creation of
the ‘magic’ code.

b) Replace the CHANCE and COMMUNITY CHEST
cards with ‘freebies’ or ‘takeaways.’ Either give
the player a free statement, or take one away.
It is important to keep the danger aspect of
MONOPOLY in your game.

c) Divide the class into small teams. The game is
played exactly like MONOPOLY, except now the
players are buying SAS code instead of
property. You can even put the major code
statements like PROC PRINT, DATA, etc. on
“Park Place” or “BoardWalk” and give them a
higher value. All MONOPOLY rules apply
(mortgaging, selling). You can’t put hotels on a
square, though.

One approach would be to have one member of each
team key in the statements on a terminal once they are
purchased. This way, the team can see which
statements or options they still need. They will also
need to agree on the right SAS code to solve the
problem.

Board games are great fun for both the instructor and
the student. They require considerable time; however,
the learning which results is immeasurable. Another
side benefit of using this training approach is stress­
reduction. Students don’t even notice they’re
learning!!
4) **Propaganda**

It isn’t accidental that politicians spend the amount of money they do on campaign buttons, pins, etc. They serve as a constant reminder. SAS training can be reinforced in very much the same way using the same techniques. Some ‘souvenirs’ that companies give training alumni are:

- buttons
- pens
- coffee cups
- ribbons

to name a few. Think of a catchy slogan to put on them that will remind them of SAS training.

For example: a button that says “SAS: Use It or Lose It” followed by your department name below can be effective.

Propaganda does require a company investment, but it can be minimal.

5) **Newsletter Challenge**

It seems to be a part of the American spirit to love a challenge. A technique I have used to facilitate SAS retention is to post a coding challenge in our quarterly newsletter. Unlike the ‘quiz’ mentioned earlier, this is meant to be difficult, and can even cover an area not taught in class. Issue the challenge in the form of a dare, such as:

“I’ll bet you can’t solve this SAS problem.”

and then present the problem. Always be sure to post the solution somewhere in the publication, though.

Not everyone will take up the challenge, but it may pique the curiosity of some of the more experienced users. You can even set it up as a contest: the first person to answer with the correct solution wins, and his/her name will be mentioned in the next newsletter.

6) **“Ten Best Ideas from Class” sheet**

This can be done in one of two ways:

a) Post the ten best ideas contributed by all students who have ever taken SAS training on a posterboard and keep it in the front of the classroom. Post their name, department, date, and idea/comment/solution.

b) Post the ten best ideas contributed by the current class. Use the same criteria as above, but you only choose comments from the current class. Post the comments in the front of the classroom, and add to them as the course progresses.

Everyone likes praise and recognition. This is especially true in a learning situation, where many individuals may feel insecure and inadequate. The posting of ideas acts as a public affirmation that the student has absorbed the course material well enough to apply it. Finally, it reinforces a positive view of the SAS training, which facilitates retention.

**ADDITIONAL SOURCES OF INFORMATION**


**Trademarks**

PC, XT, AT are registered trademarks of International Business Machines Corporation, Armonk, NY.

Post-It is a registered trademark of 3M Corporation.

SAS is a registered trademark of SAS Institute Inc., Cary, NC.