TEACHING SAS AS A FIRST COMPUTER LANGUAGE
Jean Ussery, SAS Institute Inc.

Abstract
This paper discusses Introduction to Data Processing Using SAS, the initial course in SAS/Curriculum offered by SAS Institute. It describes the basic philosophy of the course, its design, content, and flow of instruction. The paper concludes with recommendations for anyone developing a similar course.

Introduction
When the educational program of SAS Institute was established, most SAS users were experienced in using computers to manipulate data. In the past, training offered by SAS Institute began with SAS Basics, a course which sets as part of its prerequisites at least three months experience in data processing, knowledge of an operating system, and the ability to execute SAS jobs on the computer. Today, many new users do not meet these prerequisites since SAS is the first (and perhaps only) software package they learn. These new users look to the Institute to provide SAS training for everything from logging on to sophisticated techniques for data analysis.

The Institute has responded to this change in the user community with a curriculum approach to SAS training. There are three core courses in the Institute's curriculum: Introduction to Data Processing Using SAS, SAS Basics, and SAS Processing. While SAS Basics and SAS Processing are concerned with teaching the user SAS statements and techniques, the introductory course is concerned more with teaching the new user how to perform necessary activities on the computer to create, execute, and debug SAS jobs.

Design Considerations
Before being able to do much work in SAS, a user needs to be able to perform certain tasks on the computer. Although specific cases may differ, these tasks include:

1. Log on to the computer.
2. Use a text editor and appropriate system commands to create and modify files. These files may contain data or SAS code.
3. Issue a system command to associate a DDname with a raw data file.
4. Perform general housekeeping chores with system commands: erasing old files, sending files to the printer, and other activities.
5. Execute SAS jobs in one or more modes: batch, non-interactive (foreground processing), or interactive.

The basic assumption of the introductory course is that the user has no prior experience with either computer hardware or software. From that premise, the course teaches the new user how to perform the previously mentioned tasks on the system and introduces SAS programming concepts.

A minimum of SAS statements can open the door to SAS programming for the new user. Powerful and meaningful SAS jobs can be created by a user who:

1. Understands that the basic component of the SAS system is a special construction of the data known as a SAS data set.
2. Understands the concept of missing values in data, has some knowledge of when SAS might assign missing values, and knows how SAS displays missing data on output.
3. Can write a DATA step that creates a temporary SAS data set.
4. Can describe the appearance of the raw data record by writing an INPUT statement.
5. Writes assignment statements to create new variables from the input data.
6. Is able to invoke SAS procedures to work with the data.
7. Knows where to turn in the appropriate SAS documentation for additional help as needed. Can communicate knowledgeably with experienced SAS users to get help with problem-solving.

Course Flow
The introduction course teaches the novice how to accomplish the above tasks through a combination of lectures and workshops. The class is organized around six topical workshops, which are described below. (Some workshops actually consist of two sessions at the terminal.)

1. Data Input: The students use a text editor to input raw data and get accustomed to keying on the terminal and communicating with the computer.
2. Data Modification: Students use the text editor to modify computer files by adding and deleting lines and making other changes to the data.
3. System Commands: The students issue system commands to perform housekeeping chores.

4. Running SAS Jobs: The students execute existing, error-free SAS jobs that read external data, thus requiring a system command to point to the raw data file.

5. Error Detection and Correction: Students execute SAS jobs that contain errors, interpret the error messages, correct the errors, and re-execute until the program runs successfully and produces the desired output.


During the four days of the class, the students split their time between lectures and terminal sessions. Lectures explain global concepts (the "why"), while workshops illustrate the particular commands needed to complete a task (the "how"). Generally, a lecture session begins with a discussion of the previous workshop, has an explanation of some new concepts, and ends with a preview of the upcoming workshop. The course design follows the maxim: "Tell 'em what they're gonna do. Have 'em do it. Tell 'em what they did."

The sequence of instruction in the terminal sessions follows the same kind of triple exposure philosophy. In the first session, the instructions give explicit commands to the student. (Enter the PRINT command as shown.) A later workshop states each expected task in English. (Print the file.) Finally, the students are given a problem description and they must decide which commands are necessary to complete the task. (Calculate these values and produce a printed report.)

Since the assumption is that no one has any prior experience, the class begins with a discussion of the uses of computers in today's society and a survey of various types of equipment, concentrating on using a full-screen terminal to communicate to the computer system. Following the lecture is the Data Input workshop. The students key in a data file, lining up the data into specific columns with a tab option. Next, a SAS job that works on that raw data is also keyed in and stored. After the first workshop, a discussion of the function of text editors is held and a particular editor is studied. The students then move to the terminals for the Data Modification session.

Operating systems are the subject of the entire second day. Lectures deal with the concepts of communicating with the operating system and look at the general form of system commands. The System Command workshop is split into two separate terminal sessions. The students learn to:

1. list the names of files that belong to their userid,
2. browse, print, erase, and delete files from their userid,
3. copy files from a master userid,
4. allocate storage space for new files,
5. query and set system options, and
6. communicate to other users on the system.

Day three of the course begins with programming logic and uses flow diagrams as a design tool. One of these diagrams is then walked through and written in SAS code. In the workshop, the students execute a correct SAS job that they copy from a master userid. Then they execute a SAS job they keyed in during an earlier session. The afternoon contains a closer look at SAS: concepts of the DATA step and how SAS data sets are created, reading raw data with column INPUT, the log and listing files produced by SAS, and the PRINT, CONTENTS, and SORT procedures. In the follow-up terminal session, the students execute SAS jobs copied from a master userid but these jobs contain syntax errors and invalid data. These programs must be corrected until successfully executed.

On the final day, SAS topics covered include conditional processing with IF statements, controlling the variables written to the SAS data set with DROP and KEEP, controlling observations created by DELETE and subsetting IF, and a look at the MEANS and FREQ procedures. Workshops on this day have the students write and execute their own programs to solve problems.

Currently, the students are given one reading assignment on text editors. Based on comments in student course evaluations, additional reading assignments will be added to support the lectures.

Recommendations

It has probably been a long time since you knew nothing about computer processing. You've probably forgotten how difficult it was to grasp the concept of a variable (for example, NAME) and its value (for example, Jean). Remember how mind-boggling it was to try to
execute that first program? (First the ALLOCATEs, then the SAS command!) Or even how tough it was to get accustomed to using the terminal? (You mean I have to hit this ENTER key every time?)

So how do you take neophytes through the computer maze and into SAS? Hold their hands to start with, and then let go. Strive to show how simple things are. Do not razzle-dazzle new users with your skill and knowledge. Be aware that they may get frustrated easily and work to keep that frustration to a minimum. Set a tone of "I'm friendly - ask me."

Plan simple exercises and workshops but include some challenging work for the learner who moves at a faster pace.

Be prepared for the resistant learner, the one who was told by the boss, "You will take this course because I say so." Of approximately 180 students who have taken the Introduction class, at least 10 percent have entered the classroom with the "can't/won't learn it" attitude. I can honestly say that each of them has left the course with a feeling of accomplishment and success.

Recognize the usual confusion points. Know when to say to the students "It's okay to be a little confused right now - this next discussion ought to pull it together for you." Know the critical points of the class and try to get each person (or at least a majority) to that point before continuing.

Plan on spending about half of the total class time at the terminals. During a lecture, the students hear concepts explained and think they understand. However, it is at the terminal, while using those concepts to solve problems, that they assimilate the information and discover gaps in their understanding of the material.

Ideally, there should be no more than two people sharing a terminal. While it is convenient to have the terminals all located in one room, if tutors are assigned to help answer questions, the students may scatter for the workshops and then return to class.

Have the students execute most of their SAS jobs in a non-interactive mode. Beginners tend to make many keying errors which can be frustrating under interactive execution. To use the speed of interactive execution with the keying safety of batch, execute pre-written SAS code interactively with the %INCLUDE statement. (CMS users may also use the CMS GETSAS command.) When errors are detected in their SAS jobs, they can enter an editor from SAS, modify the SAS code to correct the error, and return to SAS immediately upon leaving the edit environment.

Plan on a wide range of backgrounds and prior experience. Although we advertise the Introduction course as one for new users who have no experience, some students come to the class with knowledge of other programming languages or word processing systems.

Be prepared to work with students who learn at different rates and have different levels of prior experience. Differing backgrounds can be smoothed out in the workshops by planning two or three levels for each workshop or having a minimum stopping point identified for the workshop. The faster students can go beyond the minimum to more challenging exercises while slower students are still working toward the minimum.

If at all possible, teach a "guinea pig" session of the course during the development process. Try out ideas, see what works, and discover what topics need revamping before finalizing the course content. Components of the class that deal with topics of special interest may be taught on a trial basis before the class begins.

Presentation Tips

Set a casual, relaxed atmosphere for the class - especially for the workshops. Your style as the instructor may be to clown a little, without acting childish or cute. Be nonthreatening, friendly, personable. Learn each student's name and use it often. During workshops, move around to the terminals - many people will ask questions if you are there but will not call you over. Learn to read nonverbal signals that show that a student is confused - it is usually written all over his face. Deal with little problems before they get big.

Keep the classroom discussion on an introductory level. The most common complaints of beginners are "too much material, too little time" and "over my head." Define new terminology when you use it, move at a slow pace (but do not be boring), and allow plenty of time for explanations in addition to the planned materials.

When possible, answer a question when it is asked, even if that topic is planned for a later time. If the question is beyond the scope of the course, defer it to a private discussion at the next break.

Get help during the terminal sessions. Have tutors assigned to assist during the workshops but make sure they know how to answer questions and prompt the students, not razzle-
dazzle. Be sure that they are aware of the goals of each workshop so they will not rush to show some new technique that is planned for a later time.

Conclusion

Learning to use SAS opens the door to the most powerful tool in our business environment. When the student is willing to learn, a successful learning experience depends on a well-designed set of teaching materials, delivered by an enthusiastic presenter who can communicate on an introductory level.