

The Ultimate Match Merge: Hiring the Best SAS Programmers

Neil Howard, Pfizer, Inc., Ann Arbor, MI

"The best thing you can do for your competition is to select poor quality staff."

VP Recruitment, Microsoft

Abstract

You are interviewing a candidate in a sharp navy suit, with an impressive resume and ten years of SAS experience. But, how do you tell if this ten years of experience is really ten years, or the same year over and over again? If, indeed, past behavior or past job performance is the best indicator of future performance, how is the SAS community probing for the facts? What are our assessment criteria?

The author developed a survey tool administered to selected SAS programmers, analysts, trainers, human resource experts, and managers as a reality check into our performance as interviewers and our experience as interviewees. What techniques are being used to interview and assess SAS talent?

This paper will explore the results of this survey, looking at the following interview techniques: identifying resume red flags, conducting useful phone interviews, structuring general face-to-face interviews, "measuring" intangibles, proficiency testing, and code walk-throughs. The paper includes a collection of technical SAS questions gathered from past SUGI papers, SAS users and SAS-L contributors; and discusses the development of probing technical questions, metrics, and standards.

The Survey Questionnaire

Disclaimer: The author is not an expert in survey design or analysis.

This topic was driven purely by my own curiosity and a desire to gather information on effective interviewing within the SAS community. In response to a user request for technical questions for interviews on SAS-L last year, participants submitted their favorites. The nature of the responses was inspiring. SAS programmers are at a premium; the range of talent is staggering. Yet how are we cherry-picking for our very specific needs given the kinds of questions we're asking?

Managers have complained about the lack of information available to aid them with the technical SAS component of an interview. And, I have heard comments regarding an overall ineffectiveness in assessing SAS skill level. Hence, some advice from the users themselves, with the caveat that the technical interview be conducted by

technical staff who understand the content and can evaluate the answers properly.

The survey questions and categories were derived from: 1) literature on hiring and interviewing, and 2) practical experience. The section on SAS specifics was based on the input of experienced SAS programmers, managers and SAS Institute instructors.

The survey tool itself was critiqued by several experienced SAS users and/or managers before it was administered. I should explain that the SAS questions, in particular, reflected the importance I place on understanding the internals of SAS (compile, execute, PDV), debugging and testing, and DATA step processing.

My survey didn't probe beyond essential through advanced BASE SAS skills and did not solicit information on evaluating statisticians, advanced macro programmers, AF/FSP/GRAPH/EIS gurus, and new product specialists (SAS/IntrNet, mining, warehousing, etc.) were not addressed. The survey did not include operating systems and interface issues. Open-ended questions probing for problem-solving skills and behavioral characteristics were not stressed in the survey but are addressed as a supplemental topic here.

Most reviewers felt the sections (as outlined below) were well covered. And the survey was sufficient enough to elicit good feedback and generate discussion. The results indicated how differently everyone approaches the interview process and defines a successful hire or productive programmer.

Additional questions on metrics and coding standards were added tangentially just to test the waters. The general flavor of the feedback given warrants presentation. And the overall lack of feedback indicates these are touchy, and often untouched, areas.

Survey Results: The Resume

"The closest to perfection a person ever becomes is when he fills out a job application form."

Stanley J. Randall

Resumes are like mirrors in a fun house and learning to decipher them is an art. "Providing support to Apollo scientists" may mean "coffee gopher". Ten years programming in SAS for one company may mean the same year over and over again or represent tremendous growth from running ad hoc reports to large systems design and data warehousing.

Technical inaccuracies in the resume were red flags for respondents: misspellings of hardware or software products, misuse of jargon, and misstatements in the description of applications indicate misunderstanding of usage and capabilities. Other red flag issues for respondents included gaps in employment history, fluff descriptions instead of concrete detail, and the inability to substantiate the experience claimed.

Where a given job description calls for client management where communications and presentation skills are paramount, the actual appearance and format of the resume is important to the interviewers. They look for good organization and effective explanation of technical issues. This was also important when documentation would be a large component of the job.

In general, the respondents expect a neat, accurate resume that demonstrates technical growth, career advancement, evidence of accomplishments, appropriate descriptions of technical material, and a clear understanding of technical issues (in gradients, depending on the nature of the position).

Survey Results: The Phone Interview

A telephone call is worth a thousand objections.

Most respondents reported using a telephone interview administered by technical staff to pre-screen applicants. They are looking for the essentials: number of years experience, attitude, and communications skill (telephone manner). Additionally, they wanted to determine a candidate's likes and dislikes.

Though stating it in many different ways, the respondents were looking for "chemistry" and to get a "sense of the person". Turn-offs in a telephone interview are lack of enthusiasm, bad mouthing former employers, blaming and excusing, giving affirmative answers to all questions without being able to substantiate the answer, and acting like a know-it-all.

Survey Results: The General Interview

The form and content of the face-to-face interview varies among the respondents. In general, there is a specific job description; several members of the technical team get a chance to meet the candidate. The team leads or managers do the reference checks, not non-technical human resource personnel. Interestingly, most respondents reported they had never been trained in interviewing techniques or technical interviewing.

The SAS skills of the candidate are evaluated primarily through: 1) standard (routinely used by the group) or ad hoc technical questions (driven by the resume), and 2) code samples. Most respondents had never used or received a SAS proficiency test. Even fewer had devised such a skills test for in-house use.

Since past performance is the best indicator of future performance, it is important to get this part right. Clearly, the most popular tool among the respondents is sample code; and most respondents walk through the code with the applicant to determine if: 1) they wrote the code, 2) they understand it (didn't copy it or use a shell program), 3) they can defend it, 4) they can explain why certain design methods and coding styles were used, and 5) they can explain how they know the program produces the intended results. Coding standards and evidence of adherence to SOPs are a plus in sample code.

Not all programmers are willing to bring code samples out of respect for their employer's or client's confidentiality. And there is often reuse code in large shop programs. Some fear their code will be copied. It was agreed that serious candidates may want to write some generic programs that demonstrate his capabilities.

Typical probing questions posed by the respondents detailed a problem requiring that the candidates demonstrate their approach to problem solving and how they apply analytical skills. The interviewers are not looking for a right or wrong answer, rather a demonstration of how the candidate thinks and pursues solutions, what happens when he or she gets stuck or finds problems with the data. (See Behavioral Interviewing, below.)

Survey Results: The SAS-Specific Interview

The responding interviewers typically ask questions that demonstrate an understanding of:

- ◊ data step internals (compile, execute, PDV)
- ◊ read/write of SAS and non-SAS data
- ◊ declaratives
- ◊ arrays
- ◊ fuzzy merges (IN= variables, BY groups)
- ◊ data _null_

Fewer asked questions about:

- ◊ modularity
- ◊ macro
- ◊ PROC vs. DATA step solutions
- ◊ innovative/inventive techniques/solutions.

Almost all of the respondents asked candidates to describe in detail a specific programming task they accomplished or explain a complex macro application. The objective is to find out what the programmer knows and if they know how to explain what they know.

Few respondents asked about testing, validation and debugging techniques. Those who did were fierce on the issues. Debugging and testing constitutes as much as 85% of what is considered "programming". Knowledge of syntax is just a small part; usage and understanding of functionality is important; experience is critical, and knowing how to get questions answered is even more important.

A programmer should be able to demonstrate that he understands how to gather requirements, program from them – mapping to code, and validate the results, using special features of the SAS system. Probe for their habits, methods, and attitude regarding deliverables and how they feel about program “checkers” or QC review or team walk-throughs. It is a definite plus if a candidate expresses a sincere interest in understanding the actual data or presents some evidence of having done so in the past.

Appendix A is a categorized collection of technical SAS interview questions obtained from SAS users, SAS-L contributions, and various web sites.

Survey Results: Intangibles

Although hiring is not all about “gut feelings”, a majority admitted it is a part of the equation. Interviewers are looking for certain personality traits. Not all programmers are computer science majors; many have music, social or physical science backgrounds and pursue challenging methodical, analytical activities like puzzles, games, etc. Companies are looking for professionalism tinged with a sense of humor, curiosity, self-motivation, flexibility, and creativity.

Important, too, is the ability to work in a team environment effectively, handle pressure, work against deadlines, and communicate effectively with clients. Respondents often ask for descriptions of business relationships with peers, clients, or bosses. They prefer candidates who aren't judgmental and don't act like know-it-alls. Although some hire for potential, most prefer highly motivated, interested employees. Asking good questions reflects well on an interviewee.

Could it be as simple as this? A web site spouting hiring wisdom has Kevin D. Weeks identifying the most common characteristics among superlative programmers to be:

- ⇒ Preference for cats as pets
- ⇒ Sense of the absurd
- ⇒ Fan of science fiction
- ⇒ Creativity
- ⇒ Knack for puns

Programmers are human cats: night creatures, independent, they prefer to be left alone except when they want attention, they're elegant (with code) and managing programmers has been likened to herding cats.

Programmers have a highly developed sense of the absurd to tolerate absurd schedules, absurd requirements, and absurd hours. They love science fiction and technology, especially technology that doesn't yet exist – how better to handle the constancy of change.

Many programmers are also artists: musicians, painters, or photographers, not because programming and artistic endeavors require great creativity, but because programming is more like painting than engineering. And when they make mistakes, they can code right over them.

Programming is about using language to accomplish something and programmers have a highly evolved appreciation of how a language can be manipulated to specific ends. Puns are ways of both displaying a mastery of language and honing it.

Survey Results: Metrics

A few questions regarding programming metrics and measures of productivity were included in the survey to see if anyone would rise to the bait. It is clear from the few answers that it is hard to quantify and qualify what programmers and analysts do to the satisfaction of bean counters. It is also clear from the lack of response that this is a touchy, often untouched, subject.

The managers among the respondents are often asked to devise metrics for their group. Non-technical senior management is used to seeing numbers. “Happy clients” won't satisfy this mindset; but it is equally difficult to apply metrics. Productivity measures most often mentioned are timeliness of deliverables, performance against deadlines, end-user feedback, accuracy of status reporting, ability to solve problems.

Respondents also referred to: ability to take feedback, success in teamwork, adherence to specs, self-motivation, knowing when and how to ask for help, repeat business, follow-through, willingness to share and mentor, being innovative in attacking tasks.

Several mentioned adherence to SOPs and programming standards, begging the issue of how many shops use these tools. Some use strict code standards for readability and maintainability, checklists, guidelines, and keep complete validation folders. Questions on QC and validation revealed that accuracy of results was an important, yet many did not use program checkers or QC reviewers. The measure was client acceptance.

Number of lines of code written was dismissed almost entirely as a useful metric for programmers. Comments indicate it could promote hackerism. Programmers would not be striving for the most efficient solutions and they would not be motivated to document their programs or maintain progress reports. Programmers often claim the number of hours worked is an effective measure. Managers were looking for working smarter, not longer.

One most interesting comment made on the survey indicated that reliance of metrics is used as a substitute for good technical managers who know how to manage and motivate technical employees.

The Behavioral Interview

In the spirit of “a leopard never changes his spots”, the behavioral interview operates on the premise that past behavior is the best predictor of future performance.

Typically, behavioral interview questions open with phrases like: “tell me about a time when...”, describe a

situation where...", "give me an example of...", or "tell me how you dealt with...". An interviewer will probe for work experience demonstrating such skills as: leadership, teamwork, flexibility, decision making, communication, and more. Based on the core competencies of the company or group, the questions will be structured to probe for suitable transferable behavior. Practitioners apply the S.T.A.R. model to the responses:

S = Situation
T = Task
A = Action
R = Result.

Specific descriptions of each element complete the candidate's story. A web search on "behavioral interviewing" will turn up a wealth of theory and examples for application of this technique.

A Behavioral Slant on the Technical Interview

Applying the behavioral technique to technical material is very subjective, depending on the business problems you routinely solve. And the intent would be to experience a candidate's approach to problem analysis.

Consider word problems that describe your set up for a typical or complicated merge for reporting. If input data set A contains IDNUM and EMP_NAME, data set B has IDNUM and JOB_DESC, and data set C has JOB_DESC and BASE_SAL, how would you process these data to create a report showing IDNUM and BASE_SAL?

The question could probe for an understanding of the bigger picture, such as: describe five ways of achieving efficiency in your SAS programming; what are the dependencies and tradeoffs? This would raise issues regarding hardware, human factors, technical SAS software techniques, and more.

Interesting discussions are also generated around debugging and testing habits, coding conventions, use of HELP and tech support and other technical resources.

Survey Results: Interview Extremes

"Recruiting is a triumph of hope over experience."

Respondents reported unusual experiences, including the candidate who filed her nails throughout the interview, the one who brought her badly misbehaving nine year old, the one who ate his Big Mac and french fries, the one who leaned back in the chair and put his feet on the desk, the one who brought his dog, and the one who burst into song after lunch.

Human resources lore tells of the woman who brought her parents and let them answer all the questions and the lady who wanted to change jobs to be closer to good shopping. Interviewers object to chewing gum, smoking, alcohol, excessive questions, bad-mouthing a previous boss or company, and name-dropping, among other things.

A respondent interviewer asked one person what the LAG function did, to which the candidate replied: "Is that the way you do things around here -- ask questions you already know the answers to?" Then there was the candidate whose reaction to a technical interview was: "Why are you asking me all these technical questions?"

Another respondent was very impressed by a candidate who brought SAS program samples that were color-coded. It showed good organizational skills, nice presentation, attention to detail (or perhaps too much attention to detail?).

Proficiency Testing and Certification

Several respondents reported interviewing from a "standard" list of questions devised in-house; some have been shared at previous SUGIs or on SAS-L. Others administer homegrown technical quizzes or tests to capture a candidate's technical skills. Many questions were intended to evoke an approach to a problem, without requiring a SAS code solution necessarily.

Last year, SAS-L users participated in the beta test of a certification test for SAS programmers on the web at www.tekmetrics.com. It was strictly multiple choice and drew conclusions from scores. One tester commented: "The test was somewhat interesting (but slow) to take, but also illustrated some of the problems with competence testing. There were ambiguous questions. There were poorly worded questions. There were questions that, strictly speaking, didn't have any correct answers. There were questions where not enough information was given to determine a complete answer. There were questions that appeared to have typos. There were answers which 99.9% of all SAS users will never care about." Do these tests address everything that is important about a candidate's qualifications and fit for a position?

Tekmetric's tests focus on: 1) basic programming topics: acting on selected observations, character and numeric variables, creating subsets, finding shortcuts, 2) understanding DATA step processing, using more than one observation in a calculation, dates; 3) combining SAS data sets: working with grouped or sorted observations, concatenating, merging; 4) Macro Facility: creating a macro, Macro Variables; 5) preparing data: raw data, SAS data sets; 6) PROCs: datasets, format, freq, sort, summary, means, transpose; 7) producing reports: summary tables, detail reports; 8) rules of the SAS language: formats, functions, arrays, informat; 9) storing and managing data, getting information about data sets, SAS Data Libraries, diagnosing and avoiding errors, understanding and enhancing output.

The SAS Institute this year announced a global certification program in response to industry demands for assessment tools. Exams test knowledge in core areas of SAS software programming, including data management, business intelligence reporting, and applications development. A new eight-week master course is available, in addition to regular training courses, as preparation. Experienced SAS users may need no prep.

The debates on SAS-L and panel discussions at SUGI 24 constituted early evidence that the SAS community is divided on the issue of certification. Some longtime users feel they have proven their value; beginning programmers may actually find value in obtaining job offers; high-tech companies see value-add to their reputation. You are invited to draw your own conclusions.

Networking

Another interesting phenomenon observed in interview discussions is the quality of a candidate's network. The candidate will have consistently improved their marketability by "networking" at the water cooler/coffee pot or through conference attendance. They may even have used their network in getting the interview. Some interviewees will show teaching and volunteering on the resume, an indication of practicing and honing organizational and planning skills.

Sports, professional activities, and community involvement provide more evidence of networking. Involvement, "keeping in touch", continuing education demonstrate an interest in balance and a rounded practicing of business-like skills in many venues.

The 80/20 Rule

Perhaps the simplest and best advice involved application of the 80/20 Rule. In an interview situation, ask questions 20% of the time and listen 80% to obtain the most data with which to make your hiring decisions.

Conclusions

It would not be prudent to draw conclusions from this survey, but the responses were interesting and instructive. What seems evident is that the needs of hiring entities vary greatly and the hiring process is highly subjective.

And people are looking for very different skill sets and personality traits in candidates.

What I am taking away from this exercise is a new perspective on and respect for the interview process. Armed with new information, I can hopefully contribute to the overall effectiveness of the hiring in our department.

Acknowledgments

Special thanks to Susan Fehrer, BioClin. Many thanks to Joshua Sharlin, VAST; Jeff Antonelli, Parke-Davis; Bob Harner, Robert Wood Johnson Medical School; Linda Pickle, NCI/NIH; Greg Barnes Nelson, StatProbe; as well as Lee Herman, Cindy Zender, Bob Virgile, Jodi Barnes Nelson, Andy Kuligowski, Aria Razban, Tracy Cermack, Steven Weinburg, Ian Whitlock, Heidi Markovitz, Cyndie Gareleck, Pete Hellmann, Peter Yribar, Jim Bademian, and James Gear. Thanks to the following SAS-L contributors for sharing their interview thoughts and questions with the SAS community: Andy Karp, Standish Sibley, Mike Rhodes, Paul Dorfman, Karsten Self, Jack Hamilton, Girish Patel, Timothy Berryhill, Patricia Flickner, Chris Strickland, Ian Whitlock, Erik Larsen, Bernard Tremblay, and David Nasser.

References

Deems, Richard S.: Hiring: More Than a Gut Feeling. Book-mart Press, 1995.

Messmer, Max: The Fast Forward MBA in Hiring: Finding and Keeping the Best People. John Wiley & Sons, Inc., 1998.

Yate, Martin: Hiring the Best, A Managers Guide to Effective Interviewing. Holbrook, MA: Adams Media Corporation, 1994.

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

Interview Questions

Collected from: SAS-L; SAS users, managers and instructors; individual suggestions, questions in use, etc.

NOTES:

- ⇒ the paper author does not vouch for the value of these questions
- ⇒ answers are not supplied
- ⇒ technical interviews should be conducted by qualified technical people
- ⇒ there could be multiple answers to some questions; questions may overlap
- ⇒ logic and problem solving demonstration can be more telling than a correct answer to a syntax question
- ⇒ administering non-standard proficiency tests could have legal ramifications (please visit your personnel department)
- ⇒ some great programmers are lousy interviewers/test-takers; some great interviewers/test-takers are not good programmers
- ⇒ the list of technical questions is by no means comprehensive

Very Basic

- ◊ What SAS statements would you code to read an external raw data file to a DATA step?
- ◊ How do you read in the variables that you need?
- ◊ Are you familiar with special input delimiters? How are they used?
- ◊ If reading a variable length file with fixed input, how would you prevent SAS from reading the next record if the last variable didn't have a value?
- ◊ What is the difference between an informat and a format? Name three informats or formats.
- ◊ Name and describe three SAS functions that you have used, if any?
- ◊ How would you code the criteria to restrict the output to be produced?
- ◊ What is the purpose of the trailing @? The @@? How would you use them?
- ◊ Under what circumstances would you code a SELECT construct instead of IF statements?
- ◊ What statement do you code to tell SAS that it is to write to an external file? What statement do you code to write the record to the file?
- ◊ If reading an external file to produce an external file, what is the shortcut to write that record without coding every single variable on the record?
- ◊ If you're not wanting any SAS output from a data step, how would you code the data statement to prevent SAS from producing a set?
- ◊ What is the one statement to set the criteria of data that can be coded in any step?
- ◊ Have you ever linked SAS code? If so, describe the link and any required statements used to either process the code or the step itself.
- ◊ How would you include common or reuse code to be processed along with your statements?
- ◊ When looking for data contained in a character string of 150 bytes, which function is the best to locate that data: scan, index, or indexc?
- ◊ If you have a data set that contains 100 variables, but you need only five of those, what is the code to force SAS to use only those variable?
- ◊ Code a PROC SORT on a data set containing State, District and County as the primary variables, along with several numeric variables.
- ◊ How would you delete duplicate observations?
- ◊ How would you delete observations with duplicate keys?
- ◊ How would you code a merge that will keep only the observations that have matches from both sets.
- ◊ How would you code a merge that will write the matches of both to one data set, the non-matches from the left-most data set to a second data set, and the non-matches of the right-most data set to a third data set.

Internals

- ◊ What is the Program Data Vector (PDV)? What are its functions?
- ◊ Does SAS 'Translate' (compile) or does it 'Interpret'? Explain.
- ◊ At compile time when a SAS data set is read, what items are created?
- ◊ Name statements that are recognized at compile time only?
- ◊ Identify statements whose placement in the DATA step is critical.
- ◊ Name statements that function at both compile and execution time.
- ◊ Name statements that are execution only.
- ◊ In the flow of DATA step processing, what is the first action in a typical DATA Step?
- ◊ What is _n_?

Base SAS

- ◊ What is the effect of the OPTIONS statement ERRORS=1?
- ◊ What's the difference between VAR A1 - A4 and VAR A1 - A4 ?
- ◊ What do the SAS log messages "numeric values have been converted to character" mean? What are the implications?
- ◊ Why is a STOP statement needed for the POINT= option on a SET statement?
- ◊ How do you control the number of observations and/or variables read or written?
- ◊ Approximately what date is represented by the SAS date value of 730?
- ◊ How would you remove a format that has been permanently associated with a variable??
- ◊ What does the RUN statement do?
- ◊ Why is SAS considered self-documenting?
- ◊ What areas of SAS are you most interested in?
- ◊ Briefly describe 5 ways to do a "table lookup" in SAS.
- ◊ What versions of SAS have you used (on which platforms)?

- ◊ What are some good SAS programming practices for processing very large data sets?
- ◊ What are some problems you might encounter in processing missing values?
 - * In Data steps? Arithmetic? Comparisons? Functions? Classifying data?
- ◊ How would you create a data set with 1 observation and 30 variables from a data set with 30 observations and 1 variable?
- ◊ What is the different between functions and PROCs that calculate the same simple descriptive statistics?
- ◊ If you were told to create many records from one record, show how you would do this using arrays and with PROC TRANSPOSE?
- ◊ What are `_numeric_` and `_character_` and what do they do?
- ◊ How would you create multiple observations from a single observation?
- ◊ For what purpose would you use the RETAIN statement?
- ◊ What is a method for assigning first.VAR and last.VAR to the BY group variable on unsorted data?
- ◊ What is the order of application for output data set options, input data set options and SAS statements?
- ◊ What is the order of evaluation of the comparison operators: `+` `-` `*` `/` `**` `( )` ?

Testing, debugging

- ◊ How could you generate test data with no input data?
- ◊ How do you debug and test your SAS programs?
- ◊ What can you learn from the SAS log when debugging?
- ◊ What is the purpose of `_error_` ?
- ◊ How can you put a "trace" in your program?
- ◊ Are you sensitive to code walk-throughs, peer review, or QC review?
- ◊ Have you ever used the SAS Debugger?
- ◊ What other SAS features do you use for error trapping and data validation?

Missing values

- ◊ How does SAS handle missing values in: assignment statements, functions, a merge, an update, sort order, formats, PROCs?
- ◊ How many missing values are available? When might you use them?
- ◊ How do you test for missing values?
- ◊ How are numeric and character missing values represented internally?

General

- ◊ What has been your most common programming mistake?
- ◊ What is your favorite programming language and why?
- ◊ What is your favorite operating system? Why?
- ◊ Do you observe any coding standards? What is your opinion of them?
- ◊ What percent of your program code is usually original and what percent copied and modified?
- ◊ Have you ever had to follow SOPs or programming guidelines?
- ◊ Which is worse: not testing your programs or not commenting your programs?
- ◊ Name several ways to achieve efficiency in your program. Explain trade-offs.
- ◊ What other SAS products have you used and consider yourself proficient in using?

Functions

- ◊ How do you make use of functions?
- ◊ When looking for contained in a character string of 150 bytes, which function is the best to locate that data: `scan`, `index`, or `indexc`?
- ◊ What is the significance of the 'OF' in `X=SUM(OF a1-a4, a6, a9);` ?
- ◊ What do the PUT and INPUT functions do?
- ◊ Which date function advances a date, time or date/time value by a given interval?
- ◊ What do the MOD and INT function do?
- ◊ How might you use MOD and INT on numerics to mimic SUBSTR on character strings?
- ◊ In ARRAY processing, what does the DIM function do?
- ◊ How would you determine the number of missing or nonmissing values in computations?
- ◊ What is the difference between: `x=a+b+c+d;` and `x=SUM(a,b,c,d);` ?

- ◊ There is a field containing a date. It needs to be displayed in the format "ddmonyy" if it's before 1975, "dd mon cyy" if it's after 1985, and as 'Disco Years' if it's between 1975 and 1985. How would you accomplish this in data step code? Using only PROC FORMAT.
- ◊ In the following DATA step, what is needed for 'fraction' to print to the log? data _null_; x=1/3; if x=.3333 then put 'fraction'; run;
- ◊ What is the difference between calculating the 'mean' using the mean function and PROC MEANS?

PROCs

- ◊ Have you ever used "Proc Merge"? (be prepared for surprising answers....)
- ◊ If you were given several SAS data sets you were unfamiliar with, how would you find out the variable names and formats of each dataset?
- ◊ What SAS PROCs have you used and consider yourself proficient in using?
- ◊ How would you keep SAS from overlaying the a SAS set with its sorted version?
- ◊ In PROC PRINT, can you print only variables that begin with the letter "A"?
- ◊ What are some differences between PROC SUMMARY and PROC MEANS?
- ◊ PROC FREQ:
 - * Code the tables statement for a single-level (most common) frequency.
 - * Code the tables statement to produce a multi-level frequency.
 - * Name the option to produce a frequency line items rather than a table.
 - * Produce output from a frequency. Restrict the printing of the table.
- ◊ PROC MEANS:
 - * Code a PROC MEANS that shows both summed and averaged output of the data.
 - * Code the option that will allow MEANS to include missing numeric data to be included in the report.
 - * Code the MEANS to produce output to be used later.
- ◊ Do you use PROC REPORT or PROC TABULATE? Which do you prefer? Explain.

Merging/Updating

- ◊ What happens in a one-on-one merge? When would you use one?
- ◊ How would you combine 3 or more tables with different structures?
- ◊ What is a problem with merging two data sets that have variables with the same name but different data?
- ◊ When would you choose to MERGE two data sets together and when would you SET two data sets?
- ◊ Which data set is the controlling data set in the MERGE statement?
- ◊ How do the IN= variables improve the capability of a MERGE?
- ◊ Explain the message 'MERGE HAS ONE OR MORE DATASETS WITH REPEATS OF BY VARIABLES'.

Simple statistics

- ◊ How would you generate 1000 observations from a normal distribution with a mean of 50 and standard deviation of 20. How would you use PROC CHART to look at the distribution. Describe the shape of the distribution.
- ◊ How do you generate random samples?

Customized Report Writing

- ◊ What is the purpose of the statement DATA _NULL_ ; ?
- ◊ What is the pound sign used for in the DATA _NULL_ ?
- ◊ What would you use the trailing @ sign for?
- ◊ For what purpose(s) would you use the RETURN statement?
- ◊ How would you determine how far down on a page you have printed in order to print out footnotes?
- ◊ What is the purpose of using the N=PS option?

Macro

- ◊ What system options would you use to help debug a macro?
- ◊ Describe how you would create a macro variable.
- ◊ How do you identify a macro variable?
- ◊ How do you define the end of a macro?

- ◊ How do you assign a macro variable to a SAS variable?
- ◊ For what purposes have you used SAS macros?
- ◊ What is the difference between %LOCAL and %GLOBAL?
- ◊ How long can a macro variable be? A token?
- ◊ If you use a SYMPUT in a DATA step, when and where can you use the macro variable?
- ◊ What do you code to create a macro? End one?
- ◊ Describe how you would pass data to a macro.
- ◊ You have five data sets that need to be processed identically; how would you simplify that processing with a macro?
- ◊ How would you code a macro statement to produce information on the SAS log? This statement can be coded anywhere.
- ◊ How do you add a number to a macro variable?
- ◊ If you need the value of a variable rather than the variable itself, what would you use to load the value to a macro variable?
- ◊ Can you execute a macro within a macro? Describe.
- ◊ Can you a macro within another macro? If so, how would SAS know where the current macro ended and the new one began?
- ◊ How are parameters passed to a macro?

Pharmaceutical Industry

- ◊ Describe the types of SAS programming tasks that you performed: Tables? Listings? Graphics? Ad hoc reports? Other?
- ◊ Have you been involved in editing the data or writing data queries?
- ◊ What techniques and/or PROCs do you use for tables?
- ◊ Do you prefer PROC REPORT or PROC TABULATE? Why?
- ◊ Are you involved in writing the inferential analysis plan? Tables specifications?
- ◊ What do you feel about hardcoding?
- ◊ How experienced are you with customized reporting and use of DATA _NULL_ features?
- ◊ How do you write a test plan?
- ◊ What is the difference between verification and validation?

Intangibles

- ◊ What was the last computer book you purchased? Why?
- ◊ What is your favorite all time computer book? Why?
- ◊ For contractors:
 - * Will it bother you if the guy at the next desk times the frequency and duration of your bathroom/coffee breaks on the grounds that 'you are getting paid twice as much as he is'?
 - * How will you react when, while consulting a SAS documentation manual to get an answer to a problem, someone says: 'hey, I thought you were supposed to know all that stuff already, and not have to look it up in a book!'
 - * Can you continue to write code while the rest of the people on the floor where you work have a noisy party to which you were not invited?

Non-Technical

- ◊ Can you start on Monday?
- ◊ Do you think professionally?
 - * How do you put a giraffe into the refrigerator? Correct answer: Open the refrigerator door, put the giraffe in, and close the door. This question tests whether or not the candidate is doing simple things in a complicated way.
 - * How do you put an elephant in the refrigerator? Incorrect answer: Open the refrigerator door, put in the elephant, and close the door. Correct answer: Open the refrigerator door, take out the giraffe, put in the elephant, and close the door. This question tests your foresight.
 - * The Lion King is hosting an animal conference. All the animals in the world attend except one. Which animal does not attend? Correct answer: The elephant. The elephant is in the refrigerator, remember? This tests if you are capable of comprehensive thinking.
 - * There is a river notoriously known for it's large crocodile population. With ease, how do you safely cross it? Correct answer: Simply swim across. All of the crocodiles are attending the Lion King's animal conference. This questions your reasoning ability.

Open-ended questions

- ◊ Describe a time when you were really stuck on a problem and how you solved it.
- ◊ Describe the function and utility of the most difficult SAS macro that you have written.
- ◊ Give me an example of
- ◊ Tell me how you dealt with
- ◊ How do handle working under pressure?
- ◊ Of all your work, where have you been the most successful?
- ◊ What are the best/worst aspects of your current job?
- ◊ If you could design your ideal job, what would it look like?
- ◊ How necessary is it to be creative in your work?
- ◊ If money were no object, what would you like to do?
- ◊ What would you change about your job?