

What Level am I? A Look at Categorizing a Programmer as a Beginner, Intermediate, or Advanced

Brian Varney, Experis Solutions

ABSTRACT

There are numerous times that one gets categorized by their experience level with SAS and/or as a programmer in general. Whether it be as a company hiring a programmer, a programmer determining if a presentation is appropriate for them, or a project manager building a team, it is valuable to be able to define some guidelines as to someone is a beginner, intermediate, advanced programmer. This paper intends to help this process be less subjective and error prone.

INTRODUCTION

This paper intends to help solve an age-old problem. How do you categorize a programmer into a level of beginner, intermediate, or advanced? This can help recruiters, business development managers by aiding in matching up candidates to job descriptions. This paper can also help programmers recognize their own level and possibly point out how to move from one level to the next. Finally, this can also help conference organizers categorize papers so attendees can choose the best content to pay attention to and digest. Of course, much of this is subjective and open to interpretation but hopefully this paper will help in the thought process.

PROGRAMMER ATTRIBUTES

This section delves beyond programming tools. Hands on experience with the features and functionalities of a programming language and the operating system which they will be working in is key, but first we will focus on some of the non-programming skills. We will look at that in more detail in the resume section later in this paper.

RESOURCEFULNESS

Resourcefulness is one of the most important attributes of a good programmer.

- How quickly are they able to efficiently find information that they need to be able to solve their problem?
- What resources do they take advantage of?
- Do they know how to navigate the technical support available for the technologies they program in?
- How much do they depend on team members for solving their problems?

Once someone gets past the beginner category, they should be able to be self-sufficient except for issues that are not well documented or require an approach that is not typical.

PROBLEM SOLVING APPROACHES

Problem solving approaches are usually what sets a programmer apart from the beginner category. A beginning level programmer usually is putting the most energy into just getting a successful outcome for their program. This is not always a bad situation, but they can end up with inefficient or redundant code that is hard to manage.

Programmers that are beyond the beginner level begin to develop programs that include the following attributes:

- Use metadata and data to make decisions.
- Leverage arrays, macros, and/or user defined functions to cut down on redundant code.
- Ensure their code is written to be:
 - Manageable/Maintainable
 - Dependable regardless of the data
 - Usable by others in the team
 - Efficient (resource-wise and code-wise)
- Write out pseudo code first so they can discuss it with their customer to ensure they are meeting the requirements and that the requirements make sense to them and their customer.

Advanced level programmers work like this as a habit and have a knowledge of the programming language such that they can decide between multiple approaches to a problem and choose the one best for the situation and the team they are on.

PERSONALITY

The personality of an intermediate and advanced level programmer is helpful, curious, disciplined, and process driven. They need to be able to identify and mitigate risk, so they, the customers, and team are successful. They need to have a balanced approach of solving issues using their own resources and knowing when to ask for input and help.

PROGRAMMER “DATA”

RESUMES

A resume is a great source of information in determining what experience a programmer has had in using a programming language. But it can also be deceptive. A conversation is much better as you can drill down into what they know and have done and what their approach is in problem solving.

CONVERSATION

The following sections describe some things to look for and will need to drill down on when evaluating the “level” of a programmer in a conversation.

Can they go into detail regarding the experiences on their resume?

Sometimes a person will list experiences on their resume but when you ask them about it, it turns out they were minimally involved in the project. Or they were doing very simple tasks over and over while their team members solved the hard problems? A lot of years of experience does not always equate to advanced skills.

Do they know multiple methods to solve problems?

One favorite question I ask when trying to assess the level of a programmer is “How do you determine whether to use “PROC SQL” or an approach leveraging data steps and other Base SAS procedures to solve a problem?”. Then I just listen to their response. A programmer that is an intermediate or advanced level programmer can list out which approaches are better for different type of problems.

Do they use metadata in their programming?

Using data driven techniques is a good indicator that someone is intermediate or even advanced in their programming. Do they know how to leverage the dictionary tables and/or SASHELP data sets? Can they explain how they avoid hardcoding decisions into their programs?

How have they used SAS Macro on their projects?

Some people will list macro as one of their skills but just put %macro and %mend around code without ever using SAS Macro conditional logic or SAS Macro do loops with multiple ampersand references. Do they understand the scope of a macro variable? Do they liken a SAS Macro to a sub-routine in a different language? They should understand that SAS Macro is text substitution and can behave much differently than functions or sub-routines from other languages.

Can they explain what to look for in trying to make a program more resource efficient?

Someone that really understands how SAS programs work should be able to point out several areas to look for in evaluating a program for performance. For example, do they talk about only keeping the records and variables needed as early as possible in a program. Do they mention how to use the keep, drop, and where options/statements in a SAS data step to make it work as efficiently as possible? Do they mention looking for unnecessary sorting? Maybe they mention looking for inefficient SQL joins. All are indications that they understand fundamentally how SAS works. Sometimes they will confuse memory and I/O issues.

Have they spent a majority of their career in administration versus programming?

Administration skills do not mean programming skills and vice-versa.

CONCLUSION

Categorizing a programmer as a beginner, intermediate, or advanced programmer is not an exact science. Often, someone can have advanced knowledge of one topic but beginning knowledge in another. You really need to drill down on what skills you are interested in evaluating. Very few people are advanced in every area of SAS.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Brian Varney
Experis Solutions
(269) 365-1755
brian.varney@experis.com
<https://experis.solutons>
<https://accel2r.com>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.