Up to now, we have worked with global variables only. And what I mean is, all the variables we have used have been declared in the header of our program and in our case here as well as being declared in the header, we've also declared them in the include.

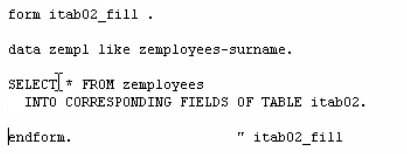
So going back down to our form, this will tell you that your form can access what are considered global variables. It can access all the variables.

Now, what we haven't discussed is the use of variables defined exclusively within our subroutine. Which we will call as a local variable. Local variables can only be accessed within the subroutine where they are declared. Once the system finishes processing code within our subroutine and control passes back to the main body of our program, all the local variables can no longer be referenced. Now some of the benefits of using local variables are that the variables only must be available in this routine, so there's no point making them available outside. So, by declaring them inside the subroutine, we save a little bit of memory usage, and because we are trying to modularize our code, by declaring local variables and using local variables, we don't have to worry about the variables being declared in the main body of our program.

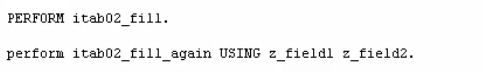
Now I mentioned a little while ago, that subroutines have an interface and we can use local variables in our subroutine to be used in the interface, and we can give them any name we want, and at the same time, the calling program, so in our case the main body.

That can use its own variable as well. It doesn't have to use the same name that we defined in the subroutine.

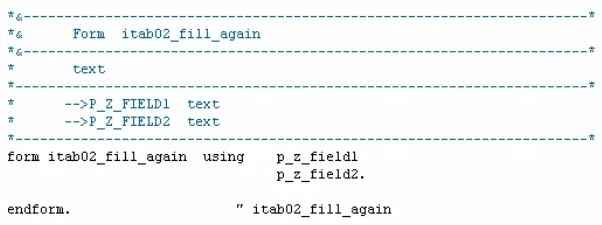
So here we have just defined a local data variable within the form. And it can only be referenced from the code we place inside the form, which we don't have much of just yet.

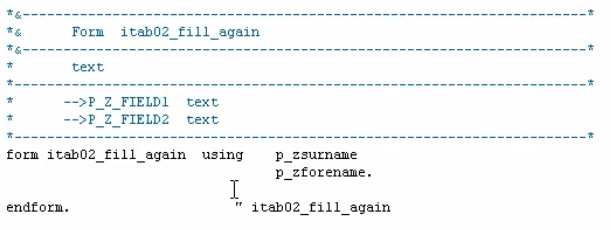


Now I mentioned we can declare variables to be used in the interface of our form. And when we do that, we're telling the system we're going to transfer data throughout the subroutine data interface.

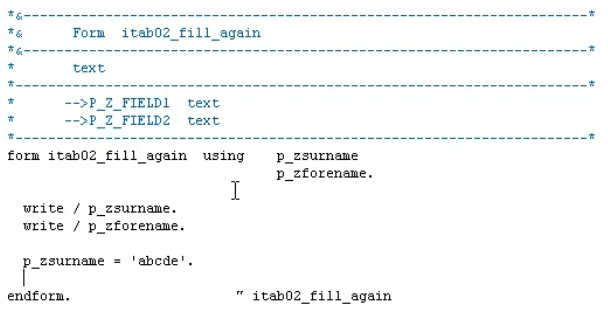




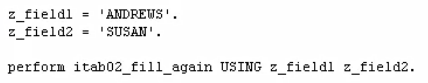


 Now, something else to note is, the form here is declared in two fields that will be used in the interface of the form but notice there's no data type. And they were taken from our perform statement. So, the fields used in our form are declared the same as those declared in the main body of the program.

When our program is executed and processing reaches the perform statements, the contents of z field 1 and z field 2 will be copied into our P Z surname and P Z forename within the form. Now one thing to bear in mind, is if we change the contents of our local fields that have been declared in the form header, so these two, you will also be changing the values of the fields that were passed into the form by our perform statement.



Now higher up in our program, where we have the perform statement, I'm just going to make sure we're going to pass in some values



So, when the perform statement is executed, these values will be passed through to the form subroutine.