

Let's look at how we implement the definition.

So, a method implementation is the actual part of the method where the functionality is placed.

Now, as you've just seen, we've defined the parameter names and parameter types in our method definition.

So, we don't list those in the implementation part at all. We don't see them. All we must do is place our code logic here.

So, what we do when we have our class implementation part of the code, we start with the class keyword. And we specified the class name and then we have implementation before we had a definition, but this part is implementation and then we list our methods out.

So here in this example, we have the method keyword. Now, this is important.

If we go back a few slides, let's go to the definition you'll see here. We use the key word methods with an S on the end, and that's for the definition.

But in the implementation, we don't use that.

so, in the implementation section, we must use the word method with no S. So, we have method then our method name.

Then we have this endmethod statement here and then between the method and endmethod, we add any logic we want.

So, as you can see with these comments, this is where you will place your code for this method.

We can make use of our var one data object.

VAR two, we can set var3 to be exported to the calling program and optionally we can change var4 that is also exported back out.

If you remember, that's the changing one. It's imported. We can change it and then we export it.

We look at method to our functional method. This is very similar and there's nothing complicated here. All we must do, if you remember how it was defined, let's scoot back.

He would go so we can see Method two was importing a variable and we are returning RetVal, so if we go back.

Here we can see in the method we don't need to do anything with the importing variable if we don't want, but we do have to set the return value for RetVal.

And here it's very simple.

We're just setting RetVal to a number one hundred twenty-three.

REPORT ZYNY\_CLASS.  
  
CLASS STUDENT DEFINITION.  
  PUBLIC SECTION.  
  DATA : NAME TYPE C LENGTH 40,  
        AGE TYPE I,  
        gender TYPE c LENGTH 1 READ-ONLY,  
        STATUS TYPE c LENGTH 1.  
\*        GENDER TYPE C LENGTH 1 READ-ONLY VALUE 'U'.  
  
  CLASS-DATA: count TYPE i.  
  
  METHODS : setname IMPORTING NAMEIN TYPE C,  
            GETNAME EXPORTING NAMEOUT TYPE C,  
            SETSTATUS CHANGING NEWSTATUS TYPE C.  
  
  PRIVATE SECTION.  
  DATA :  LOGINID TYPE C LENGTH 20,  
        PWD TYPE C LENGTH 15.  
  ENDCLASS.  
  
CLASS student IMPLEMENTATION.  
  
  METHOD setname.  
    name = namein.  
  ENDMETHOD.                    "setname  
  
  METHOD getname.  
    nameout = name.  
  ENDMETHOD.                    "getname  
  
  METHOD setstatus.  
    IF newstatus CO 'MF'." CONTAINS EITHER M OR F  
      status = newstatus.  
      newstatus = '1'.         "Set the returning value to 1 = STATUS was set  
    ELSE.  
      newstatus = '2'.       "Set the returning value to 2 = STATUS was not set  
    ENDIF.  
  ENDMETHOD.                    "setstatus  
  
ENDCLASS.                    "student IMPLEMENTATION