Subject: Problems with dummy arguments and function results
From: Van Snyder

1 Edits

Edits refer to 02-007r2. Page and line numbers are displayed in the margin. Absent other instructions, a page and line number or line number range implies all of the indicated text is to be replaced by associated text, while a page and line number followed by + (-) indicates that associated text is to be inserted after (before) the indicated line. Remarks are noted in the margin, or appear between [and] in the text.

[The phrase "its characteristics (12.2.2) are those of the function result," is backward: The characteristics of the function result are those of the result variable. It's also misplaced: The characteristics of the function result are those of the result variable, no matter whether the result variable is *result-name* or *function-name*. Editor: Delete ", its characteristics (12.2.2) are those of the function result, " at [278:1-2] and insert the sentence "The characteristics (12.2.2) of the function result are those of the result variable." before "The" at [278:5].] 278:1-2,5

[The sentence "The value of the result variable at the completion of execution of the function is the value returned by the function" is backward. It implies that the value of the result variable somehow arises from the value returned by the function. The opposite is the case. Editor: Replace it by "On completion of execution of a function, the value returned is the value of its result variable."] 278:5-6

[In the space of four lines, there are three three different ways to say that the result variable is a pointer. One is enough. There is also some confusion between present and past tense. Of the three, "is a pointer" is the shortest. Editor: At [278:6], "has the POINTER attribute" ⇒ "is a pointer"; at [278:9], "has been declared to be a pointer" ⇒ "is a pointer".] 278:6-9

2 Not sure what to do

I couldn't find anything, either in normative prose or syntax rules, that says a dummy argument is a variable. A *dummy-arg-name* isn't in the hierarchy of syntax rules starting from *variable* or *variable-name*. It seems necessary to put it under *part-name* (which is defined by the default rules), *parent-string* and *object-name*.

Although a *result-name* or *function-name* is defined to be a variable, and the circumstance under which either one is a variable is clearly defined, neither one is in the hierarchy of syntax rules starting from *variable* or *variable-name*. It seems necessary to put them under *part-name* (which is defined by the default rules), *parent-string* and *object-name*.