

Walt Brainerd's Fortran 90 programming tips

I WORKETA - March, 2004

Summary by Margarete Domingues

(www.cleanscape.net/products/fortranlint/fortran-programming_tips.html)

About Walt Brainerd

- ⑥ He has been involved with Fortran since the early 1960s.
- ⑥ Member of the ISO and ANSI committees that developed Fortran 77 and Fortran 90.
- ⑥ President of Unicomp, Inc. and "shopkeeper" of the Fortran Market.
- ⑥ Co-author of Programmer's Guide to F, The Fortran 90 Handbook, Programmer's Guide to Fortran 90, and Fortran Top 90.



***Good programming practice:
Readability and reuse tips***

General ideias

- ⑥ Include the **keyword program** and the **name of the program** in every program-end statement.
 - △ **Every variable** that is used in a Fortran program should be listed in a type declaration.
- ⑥ **Every program and procedure should include the implicit none statement** to turn off implicit typing.

(This rule may be broken for very short programs.)

- ⑥ **Echo all input data** for small amount of input data.
- ⑥ Whenever possible, **use** the default **input format**.

(This makes preparation of data easier and less prone to error).

(cont.)

- ⑥ **Warn the user** when a situation occurs that the program is not designed to handle.
- ⑥ **Always preferable clarity of the program** to shorten execution time by a few microseconds.
 - △ Not only is the program harder to get right and maintain, but with a good optimizing compiler, the improvement in execution time may be smaller than expected, or even nonexistent.
- ⑥ **Self-contained subtasks** should be written as procedures.
- ⑥ Use a **"result" variable** in every function procedure.
 - △ (It is easier to remember to use the "result" variable universally than it is to treat the "result" variable as a special case.)

Spaces and indentations

- ⑥ The **statements in each block** of a construct should be **indented** some consistent number of spaces past the statements which delimit the block.
- ⑥ **Blank characters and blank lines** should be used freely in Fortran programs.

(This improves **readability**.)

- ⑥ **Blanks should be placed before and after the percent sign** in structure component references.

(This is not required, but it improves readability.)

Declare quantities to be parameters

(whenever possible)

- ⑥ Assigning a **constant value** to a **parameter**
 - △ Compiler should produce a **diagnostic message** if the programmer inadvertently tries to change its value.
- ⑥ Use a **parameter name** instead of the corresponding constant makes a program more **readable**.
 - △ **Most important reason** for using a parameter declaration is that the program can be **modified easily** if the value represented by the parameter name needs to be changed.
 - △ The programmer can then **be sure that the constant will be correct** whenever it is used throughout the program.

DO variables

- ⑥ Use INTEGER as DO variable.

Don't use a REAL as a DO variable.

- △ Why ?

Roundoff error will cause systematic drift of successive values, and it will be impossible to guarantee that the DO block will be executed with the DO variable equal to the stopping value.

Conversion

- ⑥ When **conversion** is required **use the intrinsic type conversion functions**

(even if they are not required).

Example:

- ⑥ complex to integer or real
- ⑥ from real to integer

READ FORMAT

- ⑥ Use the **default READ FORMAT** whenever possible.
 - △ Explicit input format specifications demand **strict adherence** to specified positions for each value in the input data.
 - △ The **slightest misalignment** of the input data usually results in **incorrect values** being assigned to the variables.
 - △ By comparison, the **default** input **format** is more **tolerant** of variations in alignment, and is **easier to use**.

Dummy argument

- ⑥ Don't change the values of dummy arguments in function procedures.
- ⑥ Use an asterisk in the length declaration for a dummy argument of type character

Procedure and suffix in Fortran names

- ⑥ Use the **built-in function len** to obtain its true length when needed.
- ⑥ Always indicate the intent attribute for procedure arguments.
- ⑥ If you use Fortran name followed by the suffix type to name a derived type
 - △ Use the base name (without the suffix) for variables and structure components of that type.

For example, the component name may be type "name suffix type".



Thank you.

That's all!