

## Re: coding problem using salford FTN95

**Source:** <http://coding.derkeiler.com/Archive/Fortran/comp.lang.fortran/2004-11/0122.html>

---

**From:** Richard E Maine (*nospam\_at\_see.signature*)

**Date:** 11/03/04

Date: Tue, 02 Nov 2004 16:33:10 -0800

"bab" <bab@bab.bab> writes:

> *The problem is that he cannot make it calculate for values less than 0 but*  
> *above it is ok*

You don't say what "cannot make it calculate" means. What does he expect to happen and what actually does happen? I see nothing in the Fortran code that should particularly have a problem with values less than zero.

However, values less than zero make no physical sense here. This has nothing to do with programming. Just about the only sensible thing to do with a value less than zero for such things as a diameter is to print an error message. I would say that "not calculating" for values less than zero is exactly what I would want the program to do. I'd be more concerned that it does calculate for such numbers when ideally it shouldn't.

I do see one coding problem with the program. It wouldn't particularly cause problems for values less than zero, but it would cause problems for values of length that were not integers. These problems would be most obvious for values of length less than 1, so perhaps you meant less than 1 instead of less than zero. There is actually a problem for any non-integer value, but the problem is most obvious when the value is less than 1 (the result would be zero).

Namely, the code does not declare the types of any of its variables. Fortran lets you get by with that, but I **\*STRONGLY\*** recommend declaring the types of all variables.

Firstly, I think it really important that the programmer think about the type of every variable. It is a very important subject in programming; if you don't think at least a little about it, you will have no end of problems. Declaring the type tends to make you think at least a little about it.

Secondly, if you don't declare the types of your variables, and

don't think about the subject, you are likely to find that they have the wrong type, as presumably happened here. All the variables other than length in this code will be implicitly declared as real, which is appropriate, but length will be implicitly declared as integer, which is not a good choice.

At the very least, your friend needs to add a declaration such as

```
real :: length
```

to the code. I recommend also declaring all the other variables to be real and adding a statement

```
implicit none
```

right after the program statement. This statement will turn off the compilers implicit typing so that you will get an error message if you accidentally fail to declare the type of a variable. This helps avoid the kind of problem that this code has.

> *Program*

Not really related, but I'm surprised that the above line worked. It isn't standard Fortran. The program statement is supposed to have a program name. Doesn't much matter what the name is, but there is supposed to be one. Something like (choose any better name if you like, but this suffices for illustration)

```
Program my_program
```

```
--  
Richard Maine | Good judgment comes from experience;  
email: my first.last at org.domain | experience comes from bad judgment.  
org: nasa, domain: gov | -- Mark Twain
```