

Re: Redefining a numeric constant in Fortran (Was Re: why are some types immutable?)

Source: <http://coding.derkeiler.com/Archive/Fortran/comp.lang.fortran/2005-01/0487.html>

From: Another Jake (*anotherjake_at_anotherdoghouse*)

Date: 01/17/05

Date: Sun, 16 Jan 2005 22:33:01 -0500

When I was debugging programs about 1970 (IBM360), this was one of the first things to look for. And, then insist that the student change the argument from a number to a variable. This was a very popular error back then. I am guessing it was Fortran IV.

— AJ, FD

<beliavsky@aol.com> wrote in message
news:1105928084.551657.254220@f14g2000cwb.googlegroups.com...

> *Roy Smith wrote (in comp.lang.python):*

>

> *>Believe it or not, in some early versions of Fortran, numbers were not*

>

> *>immutable! I forget the exact scenario, but if you did something like:*

>

> *> subroutine munge (i)*

> *> i = 3*

> *> return*

>

> *>and then in your main program did:*

>

>

> *> j = 7*

> *> call munge (7)*

> *> write (6, 11) j*

> *> 11 format ('j = ', i6)*

>

> *>it would print 3! The problem is that numerical constants were*

> *interned*

> *>(i.e. in the main program, there was only a single 7 stored in*

> *memory,*

>

> *>and both uses of 7 referred to the same memory location), and the*

> *value*

comp.lang.fortran: Re: Redefining a numeric constant in Fortran (Was Re: why are some types immutable?)

> >passed to the subroutine was call by reference. It was almost as if
> the
> >compiler let you say "7 = 3" as an assignment statement.
>
> >Needless to say, people did indeed complain massively.
>
> Was this (the ability to redefine the value of '7') ever really true
of
> Fortran, or a particular Fortran compilers?
>