Subject: static variable mayhem

Posted by T Bowers on Mon, 11 May 1998 07:00:00 GMT

Hi.
I'm trying to do something very simple, but IDL 5.02 is fighting me. I'm
trying
to have what would be a static variable in c/c++ in a function set as a flag
so
that it is initialized when i come in, like so:

```
function foofunc, x
  ;In c/c++ I would put this next line to init my flag the 1st time this fn
is called
  ;  from foopro below.
  ;static int firstTimeInFooFunc = 1; ;Set to 1 ONLY the 1st time foofunc
called

  if (firstTimeInFooFunc) then begin
    firstTimeInFooFunc = 0 ;Set so this'll never happen again on subsequent
calls
    return, x = x - 1
 endif $
  else return, x = x + 1
end

pro foopro
  x = foofunc(0)
  print, x
  x = foofunc(x)
  print, x
  x = foofunc(x)
  print, x
  return
end
```

the output I want is:
-1
0
1

I tried:
```
function foofunc, x
  common CBlock, firstTimeInFooFunc = 1
...
```

but IDL won't let me initialize firstTimeInFooFunc like this.
Then I thought probably IDL would initialize it for me automatically to 0.

So i just changed the code to:

```
...
function foofunc, x
  common CBlock, firstTimeInFooFunc      ;I think this'll init to 0

  if (NOT firstTimeInFooFunc) then begin   ;init'd to 0, so I'll just NOT
the bastard
    firstTimeInFooFunc = 1 ;Set so this'll never happen again on subsequent
calls
    return, x = x - 1
 endif $
...
```

No such luck. It compiles, but when I single step to this line, I get
% Variable is undefined: FIRSTTIMEINFOOFUNC (CBLOCK).
% Execution halted at: FOOFUNC blah, blah, blah

I'm *sure* I'm missing something here. Doesn't IDL have static variables?
Can anybody help, please?