Subject: Re: _overloadMinus: what to do with invalid input?
Posted by Michael Galloy on Mon, 29 Dec 2014 19:41:23 GMT
View Forum Message <> Reply to Message

Paul van Delst <paul.vandelst@noaa.gov> wrote:

```
>  Hello,
>
>  I am overloading the "-" operator for various objects and I have a
>  philosophical question about what to do when the objects do not match.
>
>  For example, should one return a FALSE result if the objects are not the same, e.g.
>
>  FUNCTION Cloud::_overloadMinus, left, right
>    IF ( (~ ISA(left,'Cloud')) || (~ ISA(right,'Cloud')) ) THEN $
>      RETURN, FALSE
>
>  giving:
>
>>  q = fd_cloud[0] - 1
>>  help, q
>  Q               INT       =        0
>
>
>  ...or should one throw an error an halt, e.g.
>
>
>  FUNCTION Cloud::_overloadMinus, left, right
>    IF ( (~ ISA(left,'Cloud')) || (~ ISA(right,'Cloud')) ) THEN $
>      MESSAGE, 'Must supply two Cloud objects for subtraction'
>
>  leading to:
>
>>  q = fd_cloud[0] - 1
>  % CLOUD::_OVERLOADMINUS: Must supply two Cloud objects for subtraction
>
>
>  Which is the more idiomatic for IDL?
>
>  cheers,
>
>  paulv
```

I would think an error and halting, like IDL would do if you tried to use
an invalid operator with the native types. What happens if you try to add
two pointers? (not in front of my computer right now)

Mike
--

www.michaelgalloy.com
Research Mathematician
Tech-X Corporation