

Destructors and exceptions

Source: <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2004-06/0833.html>

From: David Turner (*dkturner_at_telkomsa.net*)

Date: 06/07/04

Date: 7 Jun 2004 07:51:23 -0700

Hi all

I noticed something interesting while testing some RAII concepts ported from C++ in Python. I haven't managed to find any information about it on the web, hence this post.

The problem is that when an exception is raised, the destruction of locals appears to be deferred to program exit. Am I missing something? Is this behaviour by design? If so, is there any reason for it? The only rationale I can think of is to speed up exception handling; but as this approach breaks many safe programming idioms, I see it as a poor trade.

Here is the code in question:

```
-----  
class Foo:  
    def __init__(self):  
        print "--foo %s created" % id(self)  
    def __del__(self):  
        print "--foo %s destroyed" % id(self)  
  
def normal_exit():  
    print "normal_exit starts"  
    x = Foo()  
    print "normal_exit ends"  
  
def premature_exit():  
    print "premature_exit starts"  
    x = Foo()  
    return 0  
    print "premature_exit ends"  
  
def exceptional_exit():  
    print "exceptional_exit starts"  
    x = Foo()  
    raise "oops"  
    print "exceptional_exit ends"
```

```
if __name__ == "__main__":  
    print "main block starts"  
    try:  
        normal_exit()  
        premature_exit()  
        exceptional_exit()  
    except:  
        print "exception"  
    print "main block ends"
```

The output I get is:

```
main block starts  
normal_exit starts  
--foo 141819532 created  
normal_exit ends  
--foo 141819532 destroyed  
premature_exit starts  
--foo 141819532 created  
--foo 141819532 destroyed  
exceptional_exit starts  
--foo 141819532 created  
exception  
main block ends  
--foo 141819532 destroyed
```

...which indicates to me that the destruction of the local in `exceptional_exit()` only happens when the program exits. Surely it should occur at the point at which the exception is raised?

Regards
David Turner