Ruby master - Bug #4502

strange Module.new behaviour

03/14/2011 11:35 PM - evilbrain (Andy Bogdanov)

| Status:   | Closed          |
| Priority: | Normal          |
| Assignee: | nobu (Nobuyoshi Nakada) |
| Target version: | 2.0.0 |

ruby -v:
ruby 1.9.1p430 (2010-08-16 revision 28998) [i386-mingw32]

Backport:

Description

=begin
First I create a proc:
p = proc {
  self::X = Class.new
  def self.X
    X.new
  end
}
Then I create a module:
A = Module.new(&p)
puts A.X #=> #
Then I create another module:
B = Module.new(&p)
puts B.X #=> #
Everything seems to be ok. But after module B was created (((A.X))) started to return instances of (((B::X)))
puts A.X #=> #
When (((self::X.new))) is used instead of (((X.new))) everything works as expected.

More info can be found here http://www.ruby-forum.com/topic/1275717
=end

History

#1 - 03/29/2011 05:59 PM - shyouhei (Shyouhei Urabe)
  - Status changed from Open to Assigned
  - Assignee set to ko1 (Koichi Sasada)

=end

#2 - 06/11/2011 03:52 PM - ko1 (Koichi Sasada)
  - Target version set to 2.0.0

This issue have complex problems. I'll fix it 1.9.4 or later.

#3 - 11/26/2012 09:18 AM - ko1 (Koichi Sasada)
  - Assignee changed from ko1 (Koichi Sasada) to nobu (Nobuyoshi Nakada)

nobu, could you check it?
I think you tried something like this issue.

#4 - 12/29/2012 11:46 AM - nobu (Nobuyoshi Nakada)
With latest version, A.X occurs NameError instead of returning B::X.

#5 - 01/28/2013 10:58 AM - nobu (Nobuyoshi Nakada)
  - Status changed from Assigned to Closed