

Ruby - Bug #12594

The class does not inherit from a module the modules that were included after the inclusion

07/18/2016 02:38 PM - faucct (Nikita Sokolov)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	nobu (Nobuyoshi Nakada)	
<b>Target version:</b>		
<b>ruby -v:</b>	2.3.0, 2.3.1	<b>Backport:</b> 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

**Description**

Reproduced in ruby 2.3.0 and 2.3.1  
You can see that the result changes if I swap places LimitedPeriod.include Period and Period.include Serialization.

```
module Serialization
end

module Period
end

Period.include Serialization

class LimitedPeriod
  include Period
end

LimitedPeriod < Serialization
# => true

module Serialization
end

module Period
end

class LimitedPeriod
  include Period
end

Period.include Serialization

LimitedPeriod < Serialization
# => false
```

History

#1 - 08/03/2016 10:36 AM - faucct (Nikita Sokolov)

- Description updated

#2 - 08/10/2016 04:13 AM - shyuhei (Shyouhei Urabe)

I listed this issue to be looked at the yesterday's developer meeting but sadly no time was there to deeply inspect it. At a glance it seems possible now to reflect inclusion happened afterwards. It has a chance.

#3 - 10/11/2016 10:51 AM - shyuhei (Shyouhei Urabe)

- Status changed from Open to Assigned
- Assignee set to nobu (Nobuyoshi Nakada)

We at developer meeting looked at it again. Nobu thinks he can fix this so I'm assigning to him. He said to me this is low priority, though.

#4 - 06/01/2020 10:23 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Assigned to Closed

This was implemented in [3556a834a2847e52162d1d3302d4c64390df1694](#).