Presentation 17

Classes

Announcements for This Lecture

Assignments

- **A4 Monday** at midnight
  - Hopefully you are on Task 4
  - We need to be ready to go
- Will post **A5** on **Tuesday**
  - Written assignment like A2
  - But also needs next Thurs
- Will post **A6** at same time
  - Not due until **November 15**
  - Last assignment before break

Video Lessons

- **Videos 20.1-20.8** today
- **Videos 20.9-20.10** next time
- Also **Lesson 21** next time

Exams

- No regrades after next Tues
  - Limit them to valid issues
- We will do them *eventually*

A Class Definition

```
class Example(object):


    def __init__(self,x):
        self.x = x

    def foo(self,y):
        x = self.bar(y+1)
        return x

    def bar(self,y):
        self.x = y-1
        return self.x
```

What is inside the **class folder**?

Which One is Closest to Your Answer?

A:

Example

```
__init__
foo
bar
```

B:

Example

```
foo
bar
```

C:

Example

```
__init__(self,x)
foo(self,y)
bar(self,y)
```

D:

Example

```
foo(self,y)
bar(self,y)
```

Which One is Closest to Your Answer?

A:

Example

```
__init__
foo
bar
```

B:

Example

```
foo
bar
```

C: **CORRECT**

Example

```
__init__(self,x)
foo(self,y)
bar(self,y)
```

D:

Example

```
foo(self,y)
bar(self,y)
```

A Class Definition

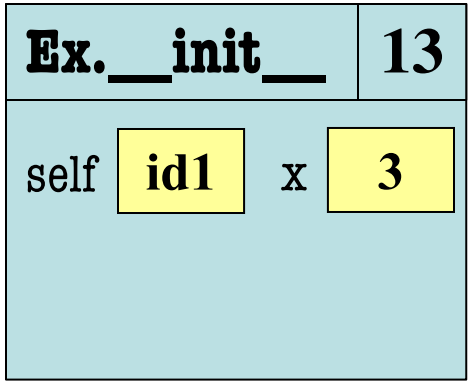
```
class Example(object):


    def __init__(self,x):
        self.x = x

    def foo(self,y):
        x = self.bar(y+1)
        return x

    def bar(self,y):
        self.x = y-1
        return self.x
```

```
>>> a = Example(3)
```

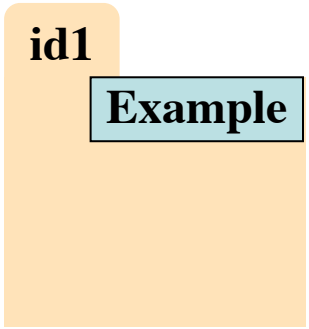
Ignoring the **class folder** what does the **call stack** and the **heap** look like?

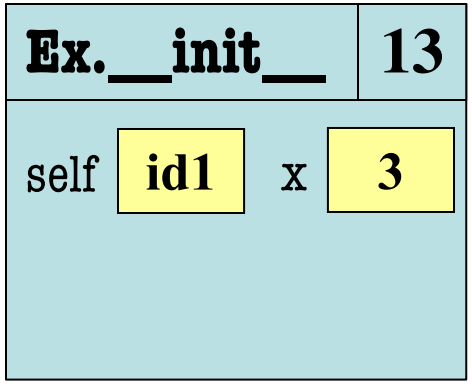
Which One is Closest to Your Answer?

A:

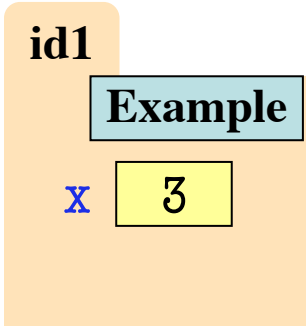


B:

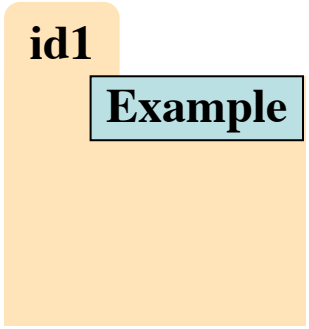




C:



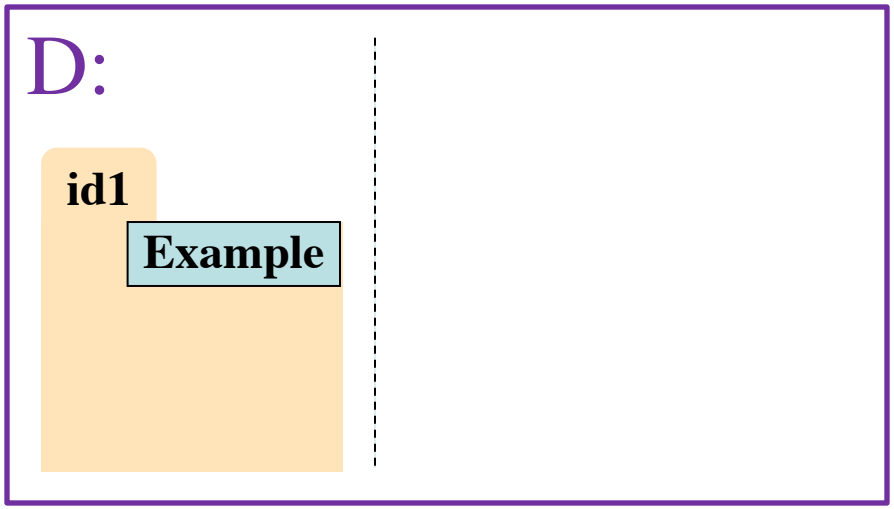
D:



A Class Definition

```
class Example(object):


    def __init__(self,x):
        self.x = x

    def foo(self,y):
        x = self.bar(y+1)
        return x

    def bar(self,y):
        self.x = y-1
        return self.x
```

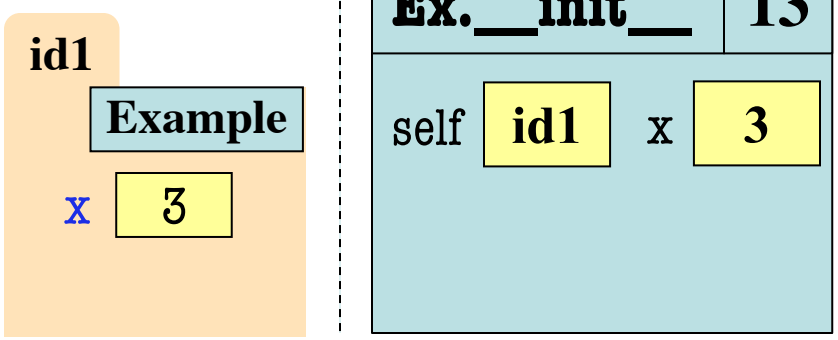
```
>>> a = Example(3)
```



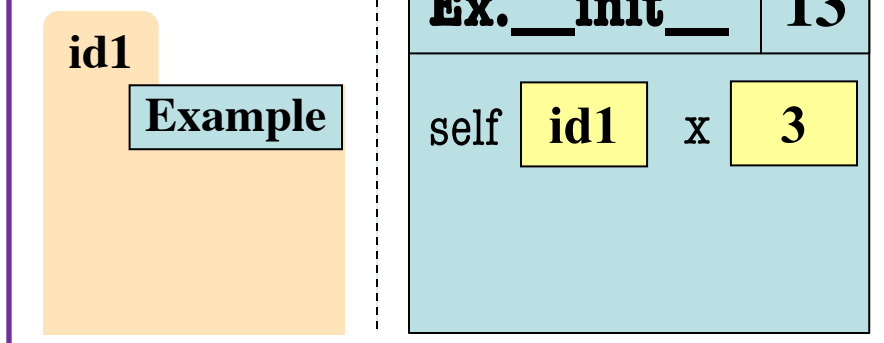
What is the **next step**?

Which One is Closest to Your Answer?

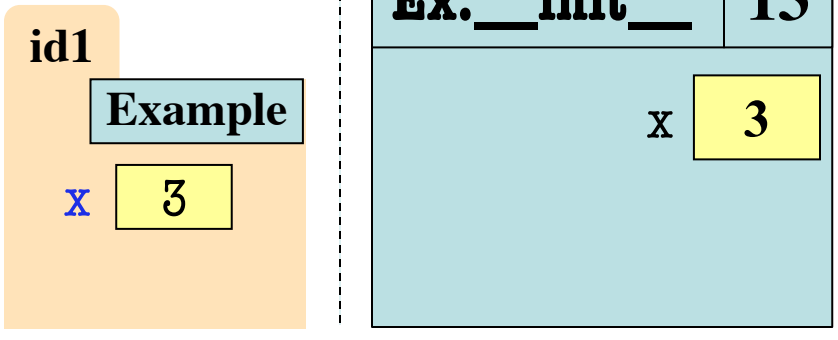
A:



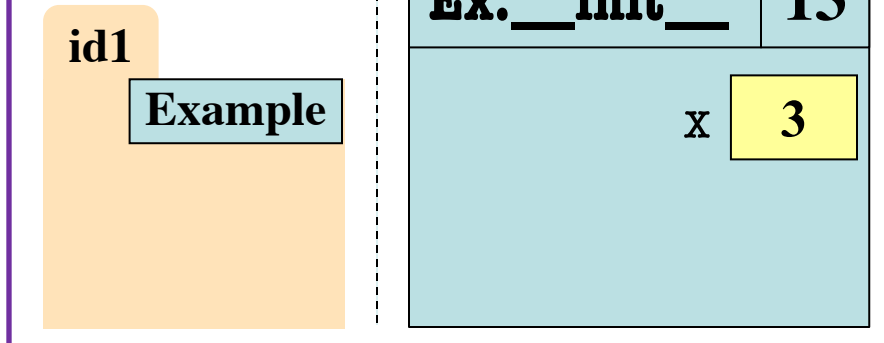
B:



C:



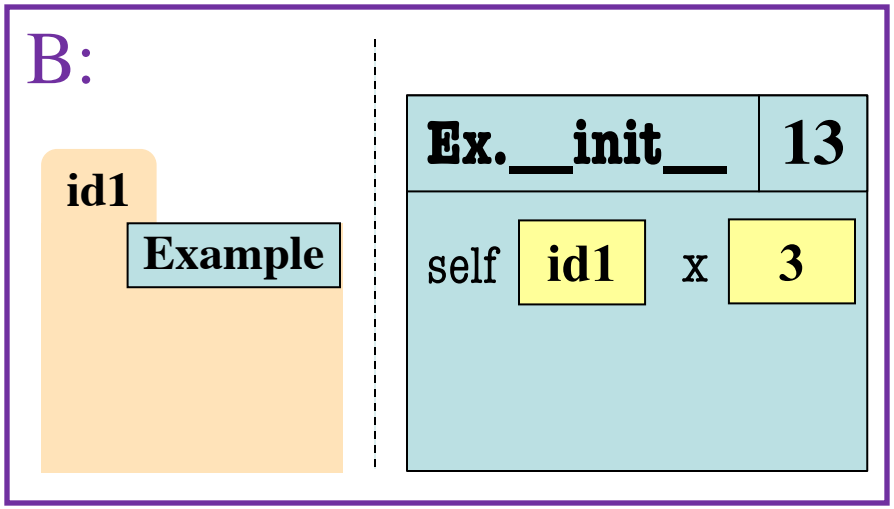
D:



A Class Definition

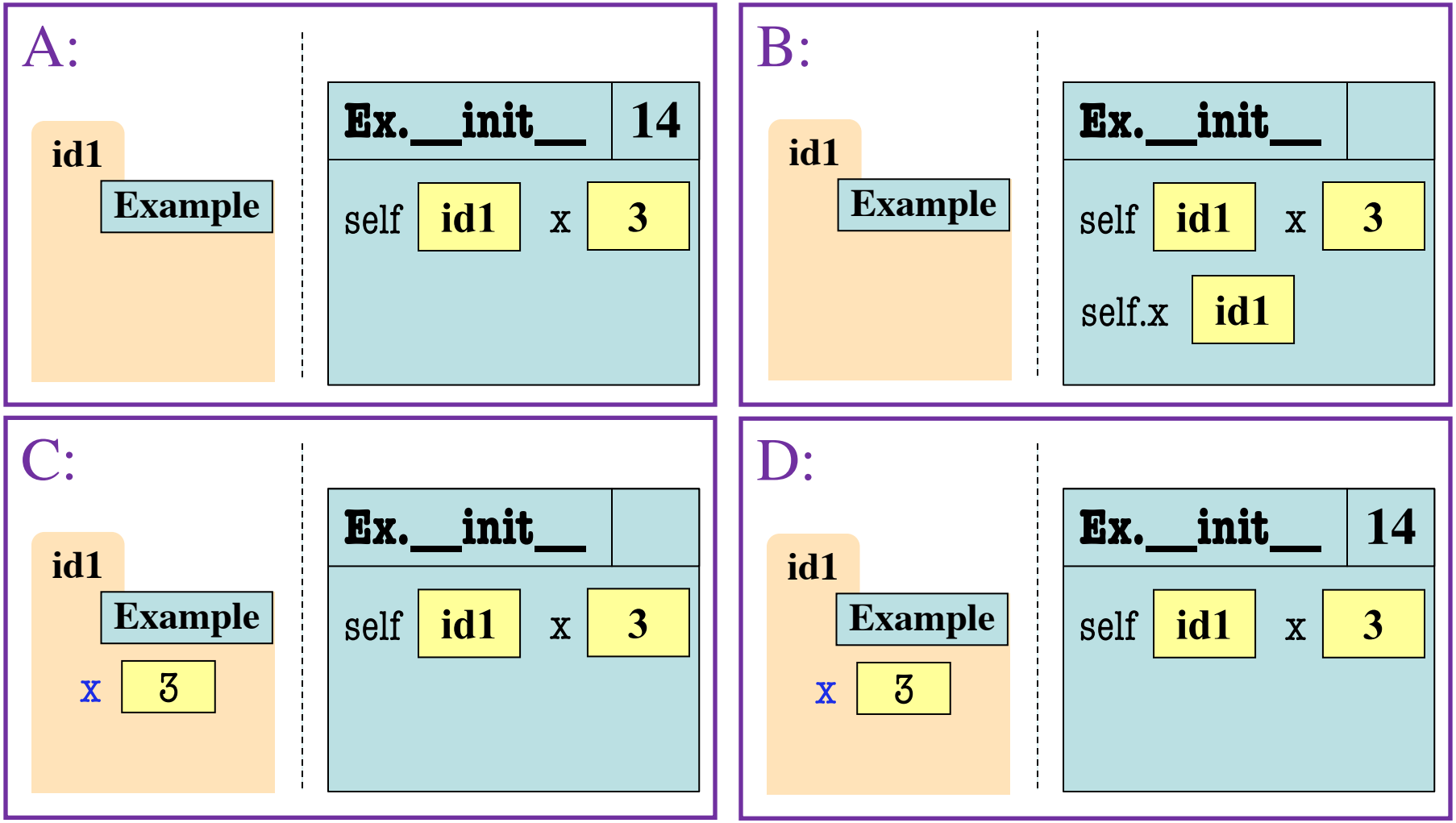
```
class Example(object):


    def __init__(self,x):
        self.x = x

    def foo(self,y):
        x = self.bar(y+1)
        return x

    def bar(self,y):
        self.x = y-1
        return self.x
```

```
>>> a = Example(3)
```



What is the **next step**?

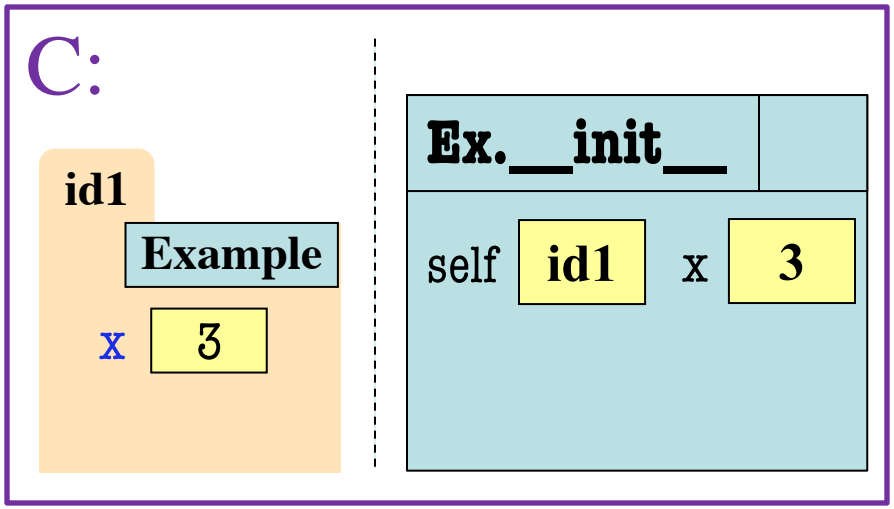
Which One is Closest to Your Answer?



A Class Definition

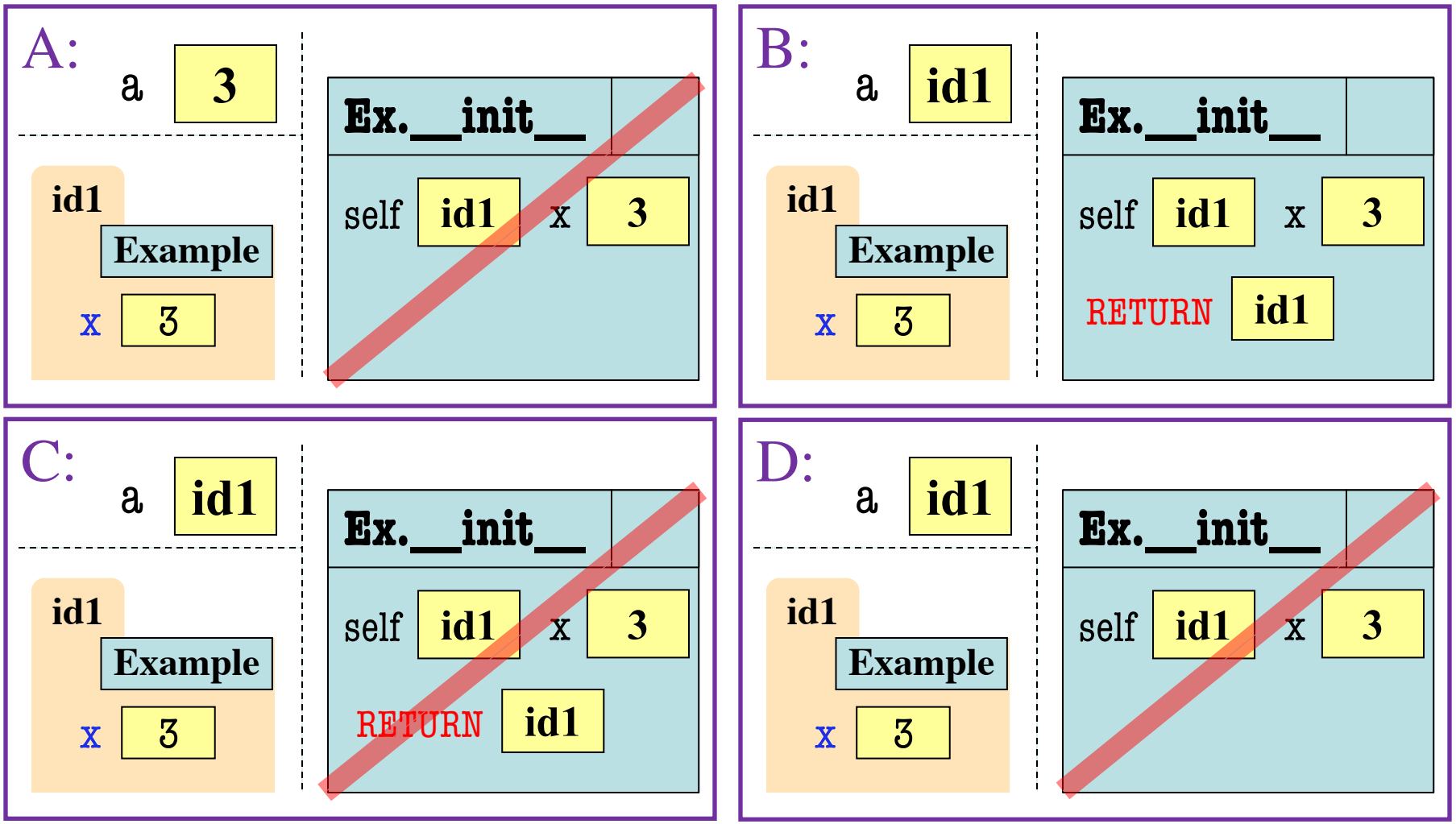
```
class Example(object):


    def __init__(self,x):
        self.x = x

    def foo(self,y):
        x = self.bar(y+1)
        return x

    def bar(self,y):
        self.x = y-1
        return self.x
```

```
>>> a = Example(3)
```



What is the **next step**?

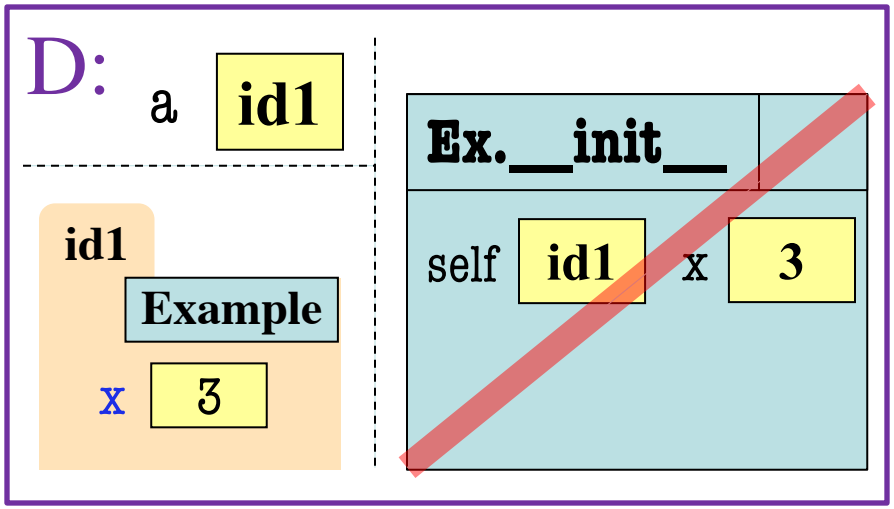
Which One is Closest to Your Answer?



A Class Definition

```
class Example(object):


    def __init__(self,x):
        self.x = x

    def foo(self,y):
        x = self.bar(y+1)
        return x

    def bar(self,y):
        self.x = y-1
        return self.x
```

```
>>> a = Example(3)
```



Initializer has no return

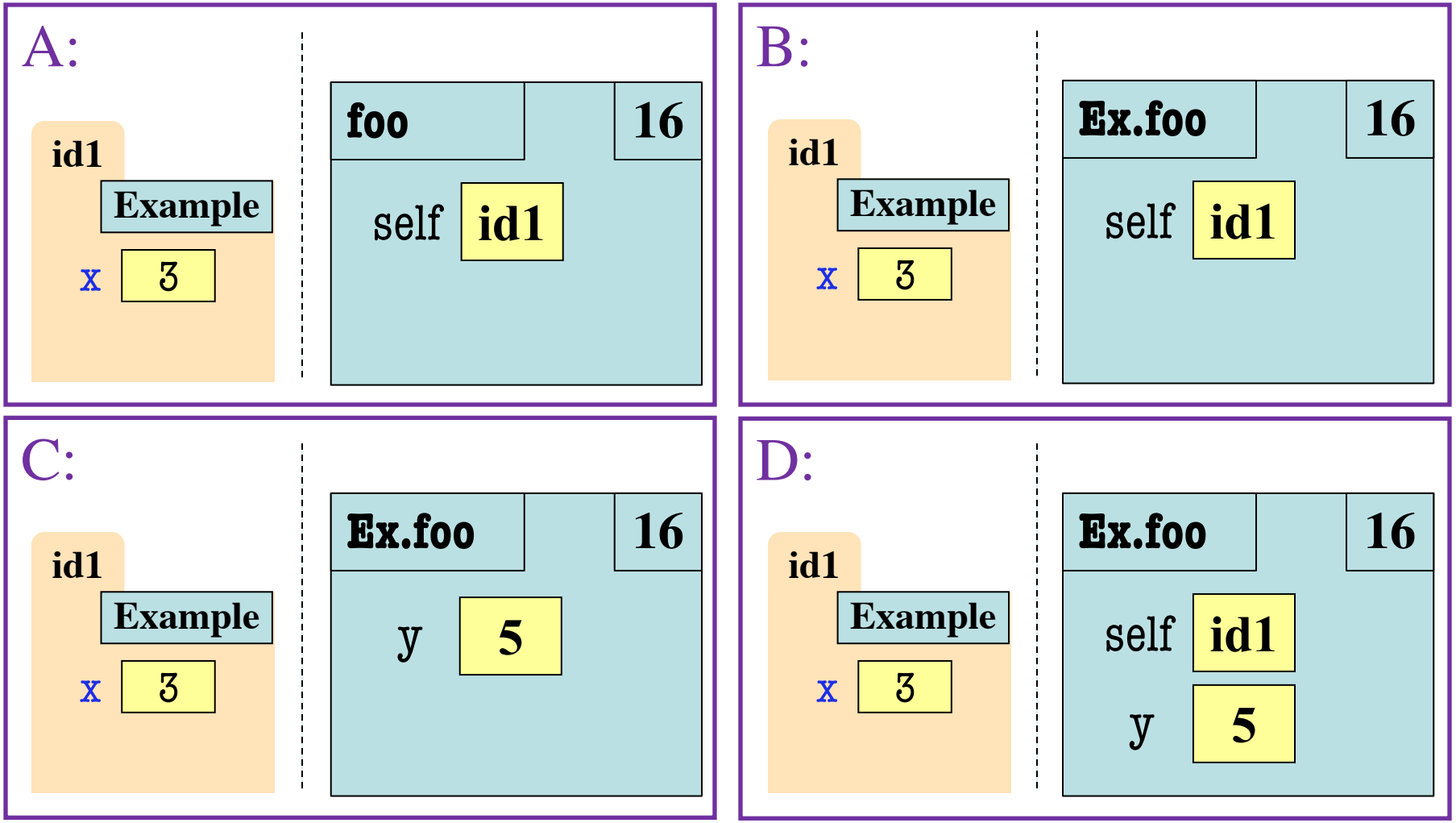
A Class Definition

```
   class Example(object):
       
       
12     def __init__(self,x):
13         self.x = x
14
15     def foo(self,y):
16         x = self.bar(y+1)
17         return x
18
19     def bar(self,y):
20         self.x = y-1
21         return self.x
```

```
>>> a = Example(3)
>>> z = a.foo(5)
```

Ignoring the **class folder** what does the **call stack** and the **heap** look like?

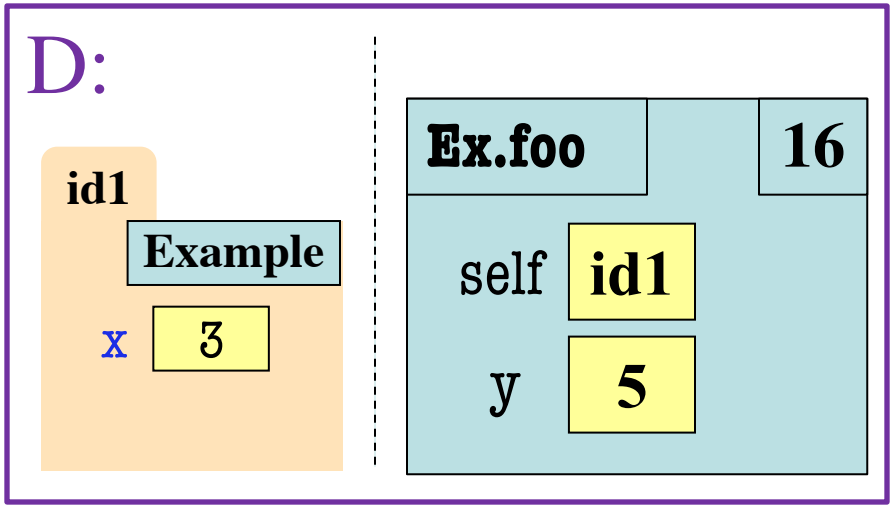
Which One is Closest to Your Answer?



A Class Definition

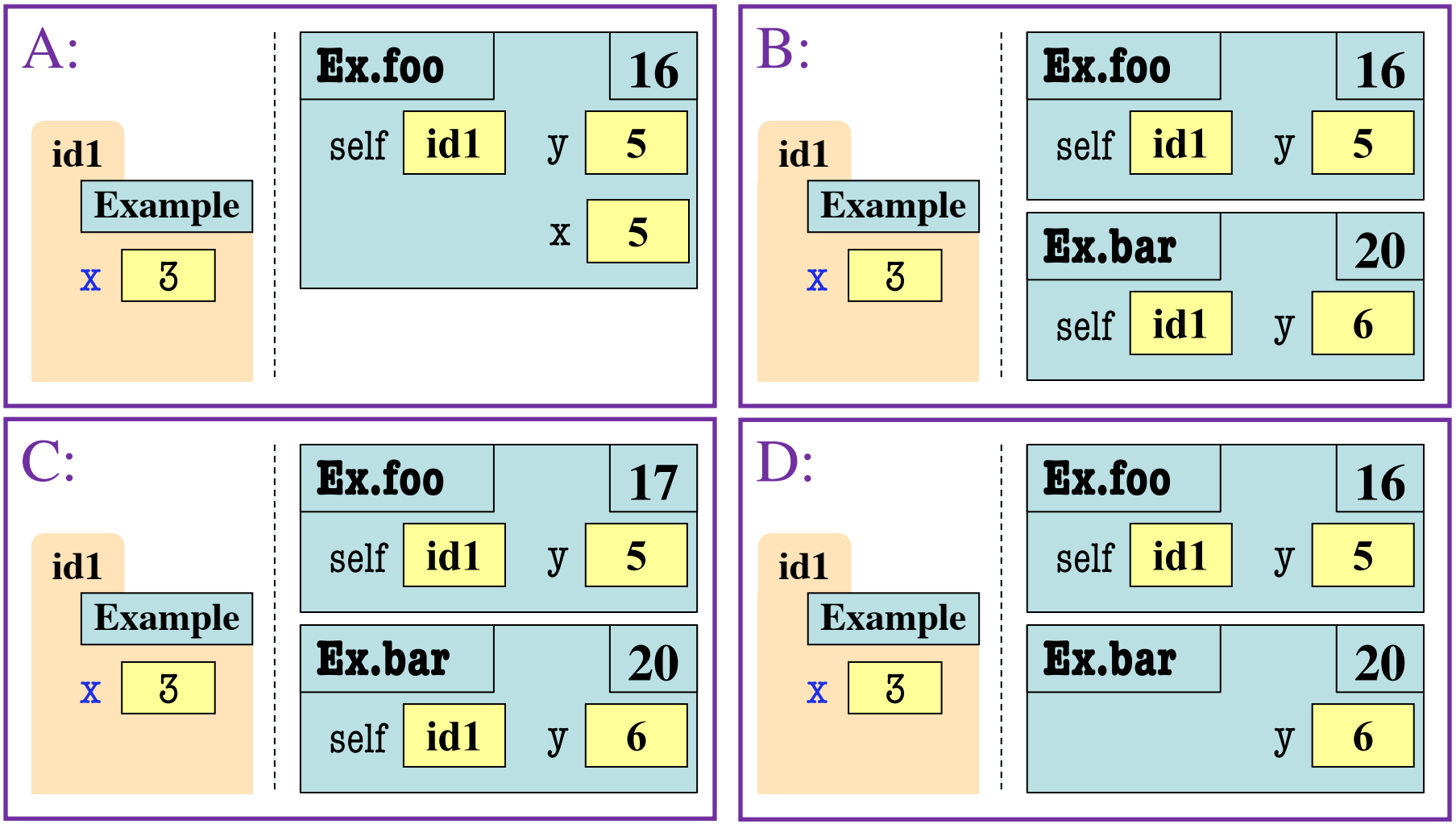
```
   class Example(object):
       
       
12     def __init__(self,x):
13         self.x = x
14
15     def foo(self,y):
16         x = self.bar(y+1)
17         return x
18
19     def bar(self,y):
20         self.x = y-1
21         return self.x
```

```
>>> a = Example(3)
>>> z = a.foo(5)
```



What is the **next step**?

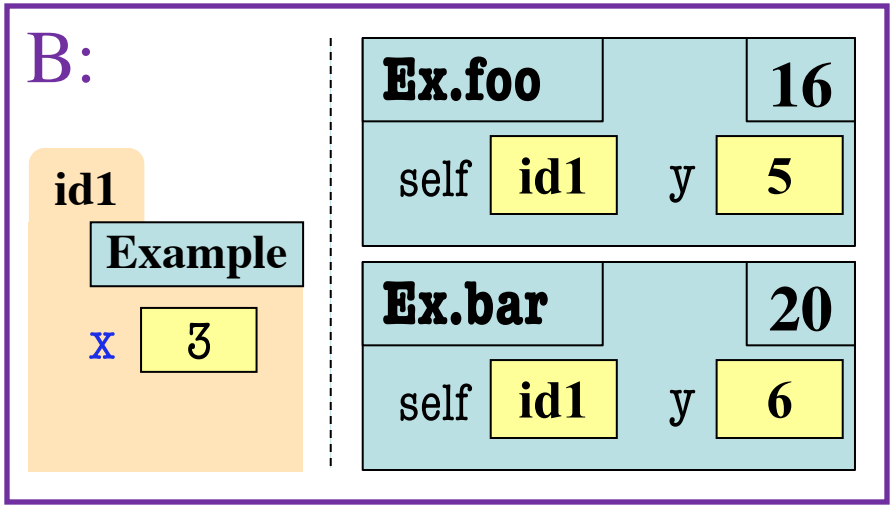
Which One is Closest to Your Answer?



A Class Definition

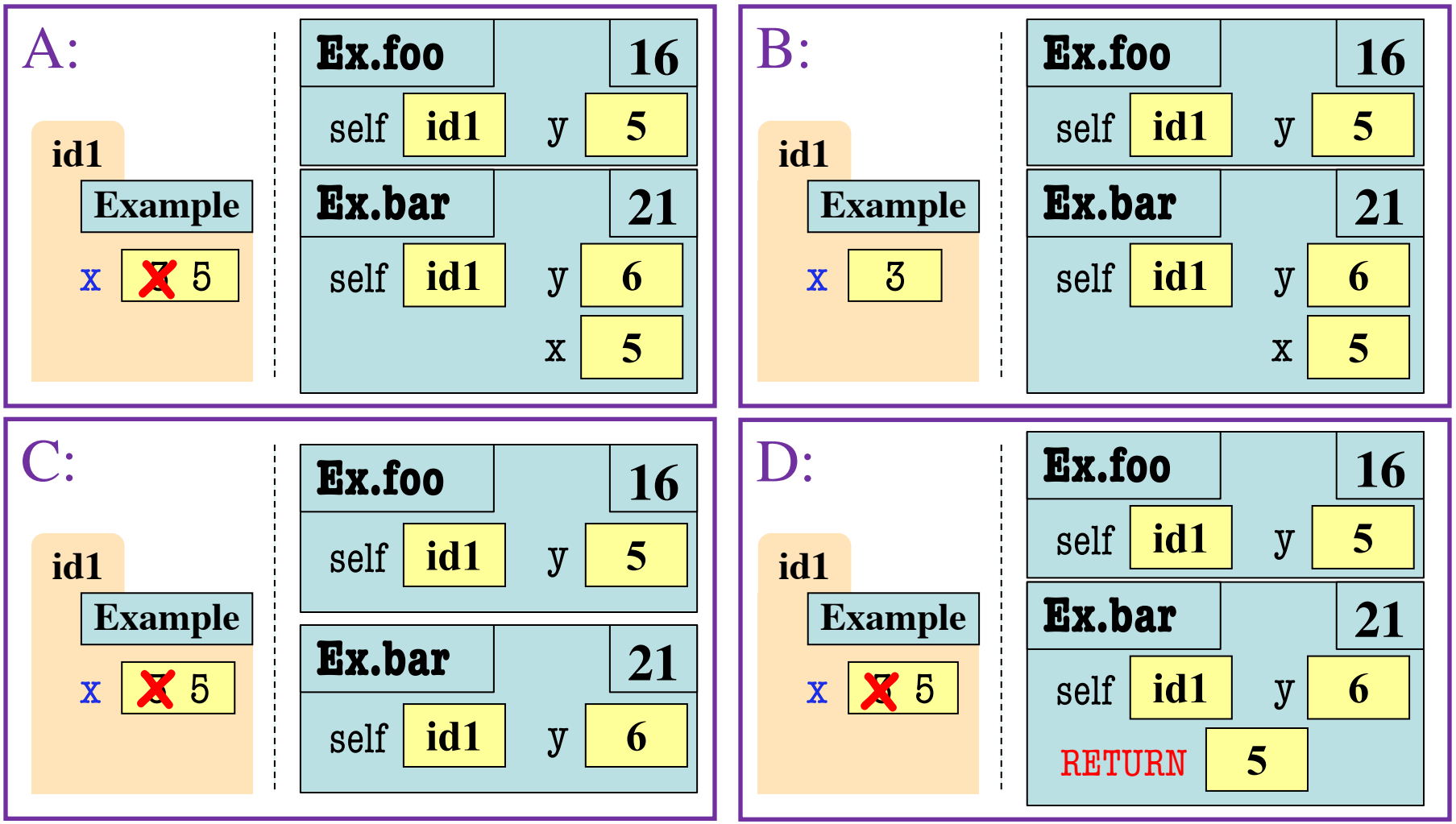
```
   class Example(object):
       
12     def __init__(self,x):
13         self.x = x
14
15     def foo(self,y):
16         x = self.bar(y+1)
17         return x
18
19     def bar(self,y):
20         self.x = y-1
21         return self.x
```

```
>>> a = Example(3)
>>> z = a.foo(5)
```



What is the **next step**?

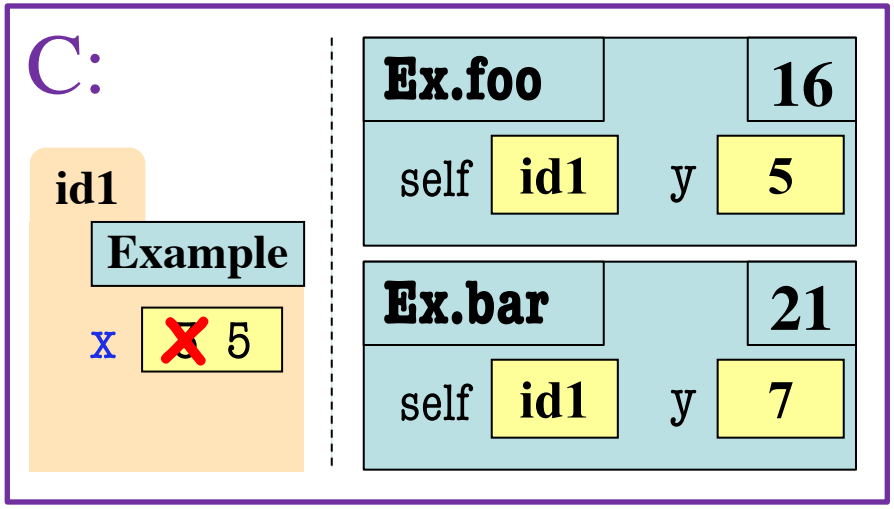
Which One is Closest to Your Answer?



A Class Definition

```
   class Example(object):
       
       
12     def __init__(self,x):
13         self.x = x
14
15     def foo(self,y):
16         x = self.bar(y+1)
17         return x
18
19     def bar(self,y):
20         self.x = y-1
21         return self.x
```

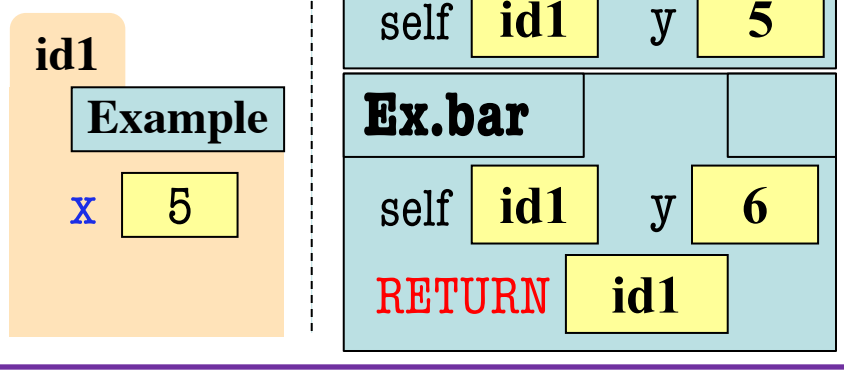
```
>>> a = Example(3)
>>> z = a.foo(5)
```



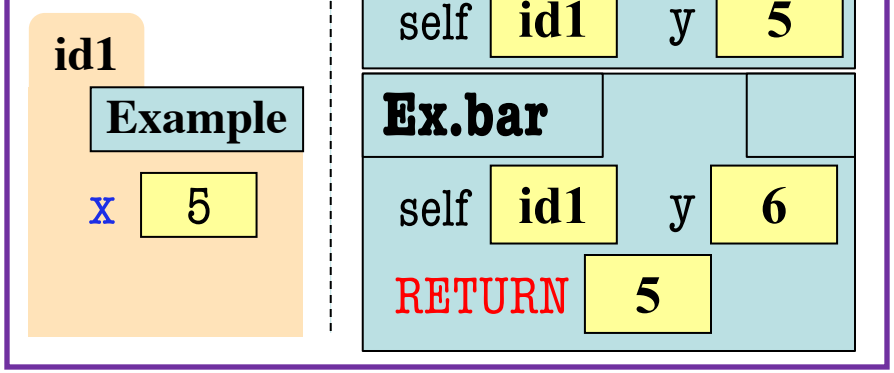
What is the **next step**?

Which One is Closest to Your Answer?

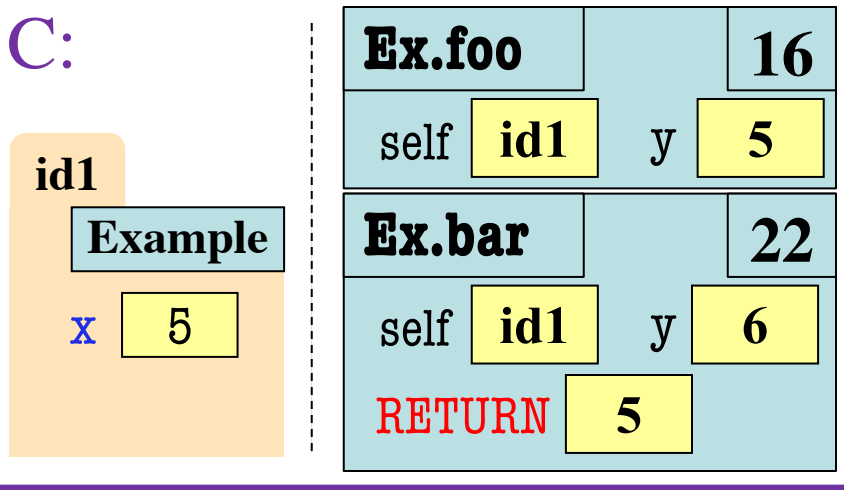
A:



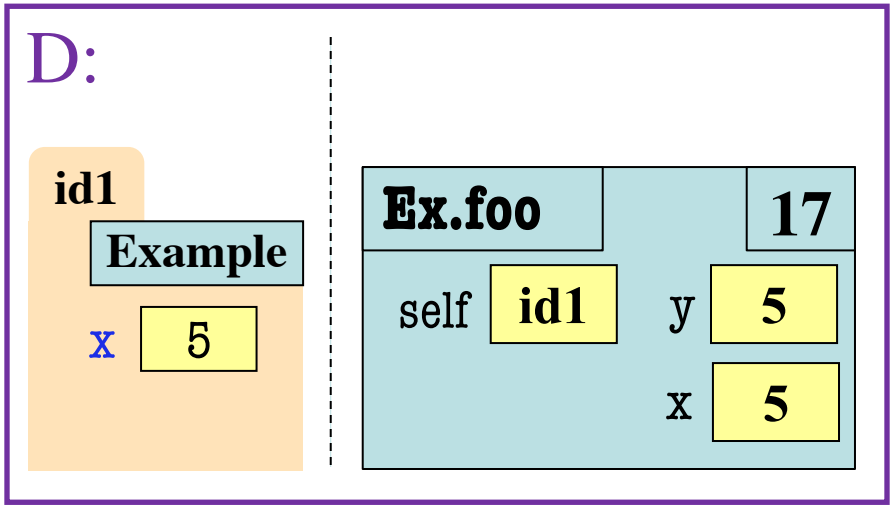
B:



C:



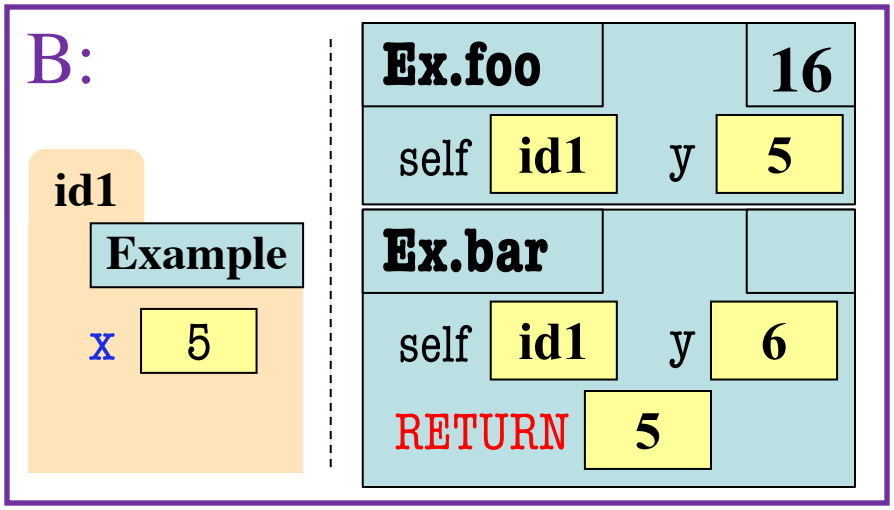
D:



A Class Definition

```
   class Example(object):
       
       
12     def __init__(self,x):
13         self.x = x
14
15     def foo(self,y):
16         x = self.bar(y+1)
17         return x
18
19     def bar(self,y):
20         self.x = y-1
21         return self.x
```

```
>>> a = Example(3)
>>> z = a.foo(5)
```



Try the rest on your own

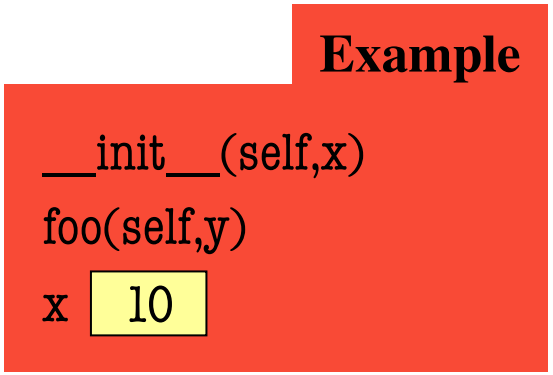
Another Class Definition

```
class Example(object):


    x = 10

    def __init__(self,x):
        self.x = x
        self.foo(x+1)

    def foo(self,y):
        self.y = y
```

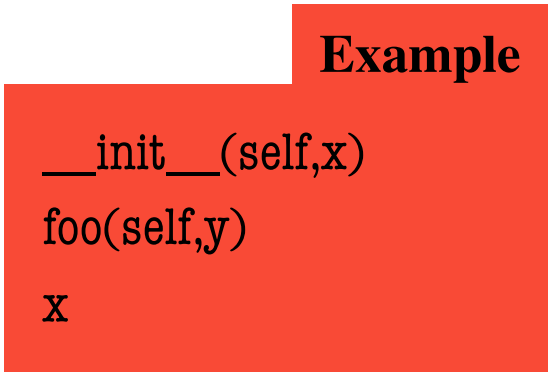
What is inside the **class folder**?

Which One is Closest to Your Answer?

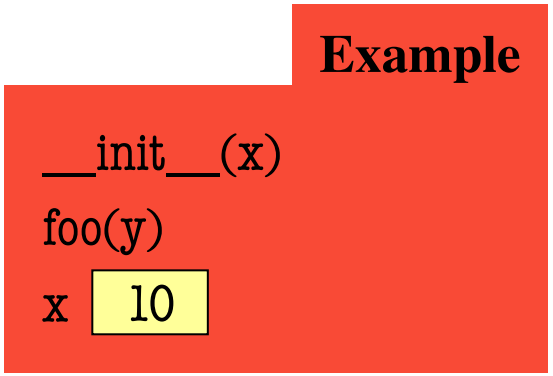
A:



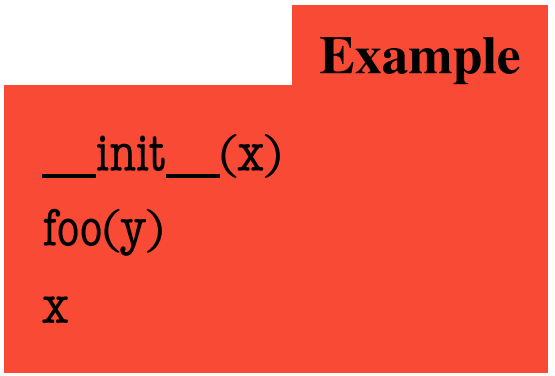
B:



C:



D:



Which One is Closest to Your Answer?

A: **CORRECT**

Example

```
__init__(self,x)
foo(self.y)
x  10
```

B:

Example

```
__init__(self,x)
foo(self.y)
x
```

C:

Example

```
__init__(x)
foo(y)
x  10
```

D:

Example

```
__init__(x)
foo(y)
x
```

Another Class Definition

```
class Example(object):


    x = 10

    def __init__(self,x):
        self.x = x
        self.foo(x+1)

    def foo(self,y):
        self.y = y
```

```
>>> a = Example(3)
```

Challenge:
Visualize this call

Making A New Class

```
class Line(object):
    """A class representing a line segment.

    A line segment is defined by two (2d) points: the starting point and
    the ending point.

    Attributes x1, y1: The start point
    Invariant: x1, y1 are floats

    Attribute x2, y2: The ending point
    Invariant: x2, y2 are floats
    """
```

The Initializer

```
define __init__(???):
    """
    Initializes a new line segment

    Parameter x1: The x-coordina
    Precondition: ????

    Parameter y1: The y-coordina
    Precondition: ????
    ... (see line.py)
    """
    pass
```

What are params?

A: (x1,y1,x2,y2)

B: (self,x1,y1,x2,y2)

C: (self,p1,p2)

D: (p1,p2)

E: Unsure

The Initializer

```
define __init__(???):
    """
    Initializes a new line segment

    Parameter x1: The x-coordina
    Precondition: ????

    Parameter y1: The y-coordina
    Precondition: ????
    ... (see line.py)
    """
    pass
```

What are params?

A: (x1,y1,x2,y2)

B: (self,x1,y1,x2,y2)

C: (self,p1,p2)

D: (p1,p2)

E: Unsure

A Method

```
def length(???):
    """
    Returns the length of this line
    """
    pass
```

What are params?

A: (x1,y1,x2,y2)

B: (self,x1,y1,x2,y2)

C: ()

D: (self)

E: Unsure

A Method

```
def length(???):
    """
    Returns the length of this line
    """
    pass
```

What are params?

A: (x1,y1,x2,y2)

B: (self,x1,y1,x2,y2)

C: ()

D: (self)

E: Unsure

Another Method

```
def shift(???):
    """
    Shifts this line by the given amount

    Parameter dx: The amount to shift the x-coordinates
    Precondition: dx is a float

    Parameter dy: The amount to shift the x-coordinates
    Precondition: dy is a float
    """
    pass
```

Questions?