

# 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):  
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
```

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):
```

```
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
```

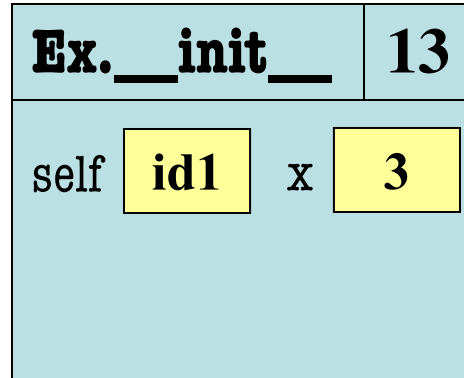
```
21  |   return self.x
```

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

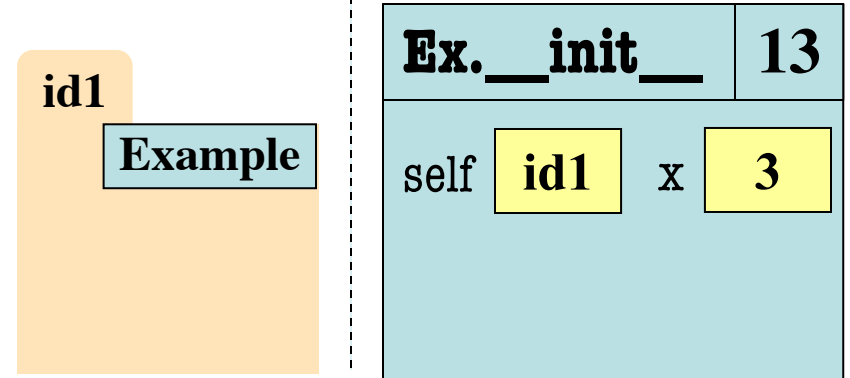
**Ignoring** the **class folder**  
what does the **call stack**  
and the **heap** look like?

# Which One is Closest to Your Answer?

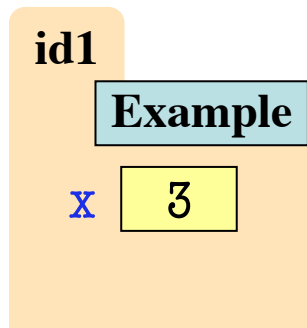
A:



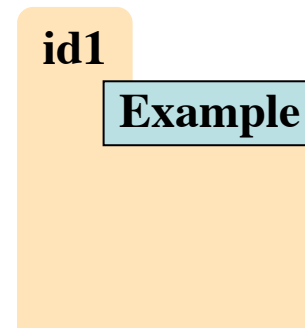
B:



C:



D:



# A Class Definition

```
class Example(object):
```

```
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)
```

D:

id1

Example

What is the **next step**?



# Which One is Closest to Your Answer?

**A:**

Diagram A illustrates the relationship between a variable `id1` and a function `Ex.__init__`. On the left, an orange box labeled `id1` contains a light blue box labeled `Example` and a yellow box labeled `3` with a blue `x` pointing to it. On the right, a light blue box labeled `Ex.__init__` contains a yellow box labeled `13` and a yellow box labeled `3` with a blue `x` pointing to it. A dashed vertical line separates the two boxes.

**B:**

Diagram B illustrates the relationship between a variable `id1` and a function `Ex.__init__`. On the left, an orange box labeled `id1` contains a light blue box labeled `Example`. On the right, a light blue box labeled `Ex.__init__` contains a yellow box labeled `13` and a yellow box labeled `3` with a blue `x` pointing to it. A dashed vertical line separates the two boxes.

**C:**

Diagram C illustrates the relationship between a variable `id1` and a function `Ex.__init__`. On the left, an orange box labeled `id1` contains a light blue box labeled `Example` and a yellow box labeled `3` with a blue `x` pointing to it. On the right, a light blue box labeled `Ex.__init__` contains a yellow box labeled `13` and a yellow box labeled `3` with a blue `x` pointing to it. A dashed vertical line separates the two boxes.

**D:**

Diagram D illustrates the relationship between a variable `id1` and a function `Ex.__init__`. On the left, an orange box labeled `id1` contains a light blue box labeled `Example`. On the right, a light blue box labeled `Ex.__init__` contains a yellow box labeled `13` and a yellow box labeled `3` with a blue `x` pointing to it. A dashed vertical line separates the two boxes.

# 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
```

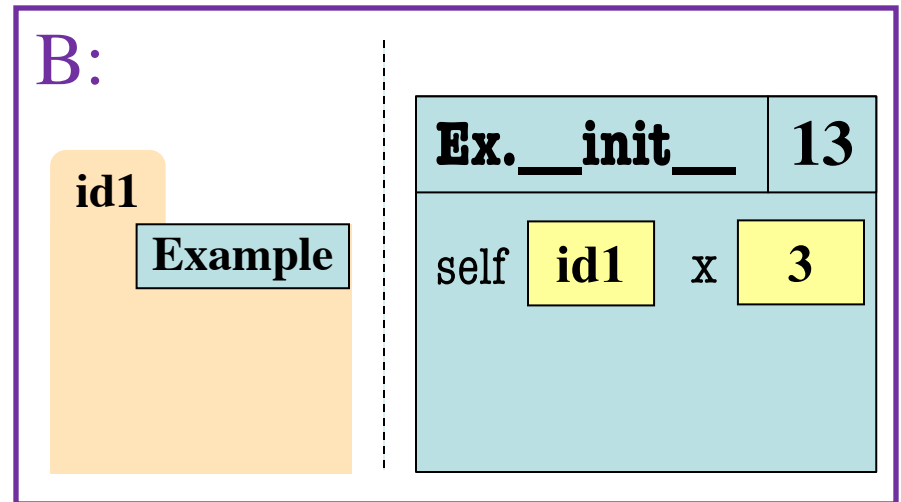
```
18 |
```

```
19 def bar(self,y):
```

```
20 |     self.x = y-1
```

```
21 |     return self.x
```

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



What is the **next step**?

# Which One is Closest to Your Answer?

A:

id1

Example

<b>Ex.__init__</b>	14
--------------------	----

self	id1	x	3
------	-----	---	---

B:

id1

Example

<b>Ex.__init__</b>	
--------------------	--

self	id1	x	3
------	-----	---	---

self.x	id1
--------	-----

C:

id1

Example

x

3

<b>Ex.__init__</b>	
--------------------	--

self	id1	x	3
------	-----	---	---

D:

id1

Example

x

3

<b>Ex.__init__</b>	14
--------------------	----

self	id1	x	3
------	-----	---	---

# 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
```

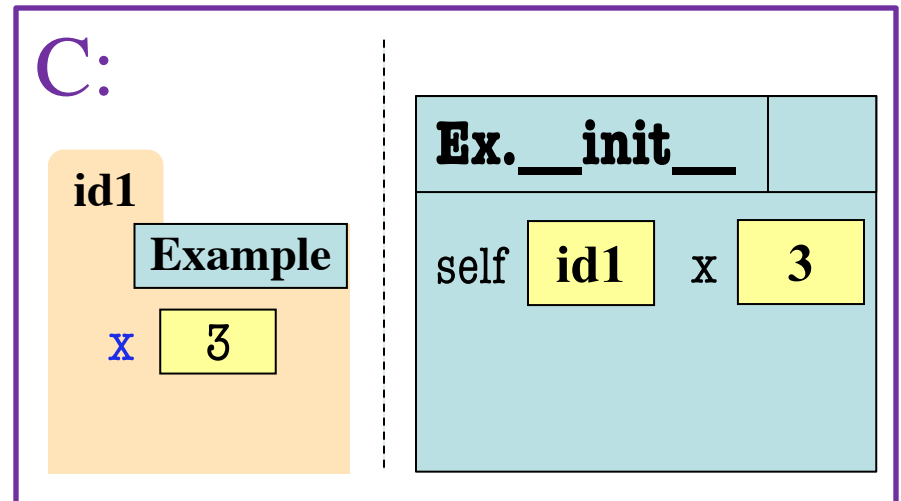
```
18 |
```

```
19 def bar(self,y):
```

```
20 |     self.x = y-1
```

```
21 |     return self.x
```

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



What is the **next step**?

# Which One is Closest to Your Answer?

**A:**

a 3

---

id1

Example

x 3

**Ex.\_\_init\_\_**

self id1 x 3

**B:**

a id1

---

id1

Example

x 3

**Ex.\_\_init\_\_**

self id1 x 3

**RETURN** id1

**C:**

a id1

---

id1

Example

x 3

**Ex.\_\_init\_\_**

self id1 x 3

**RETURN** id1

**D:**

a id1

---

id1

Example

x 3

**Ex.\_\_init\_\_**

self id1 x 3

# 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
```

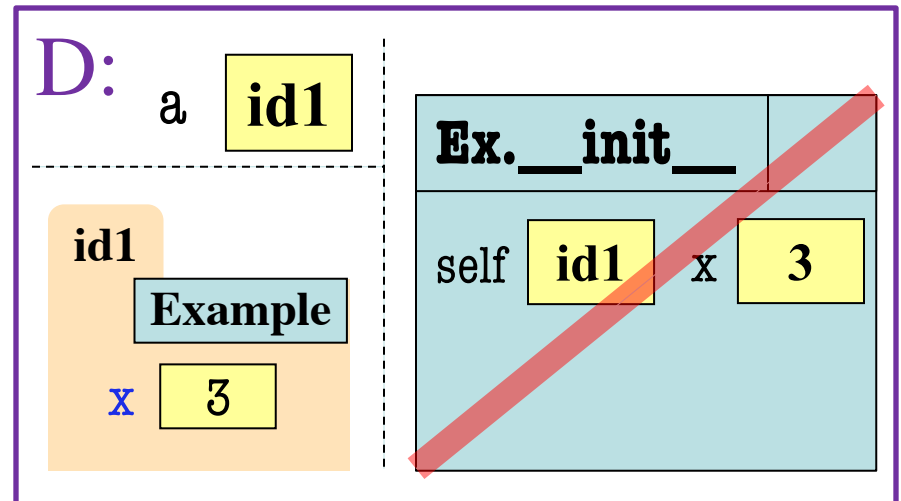
```
18
```

```
19 def bar(self,y):
```

```
20 |     self.x = y-1
```

```
21 |     return self.x
```

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



Initializer has no return

# A Class Definition

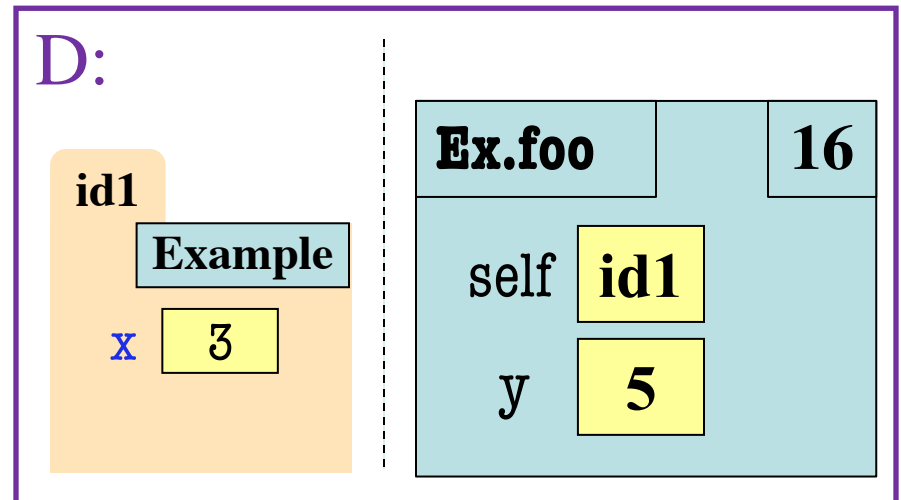
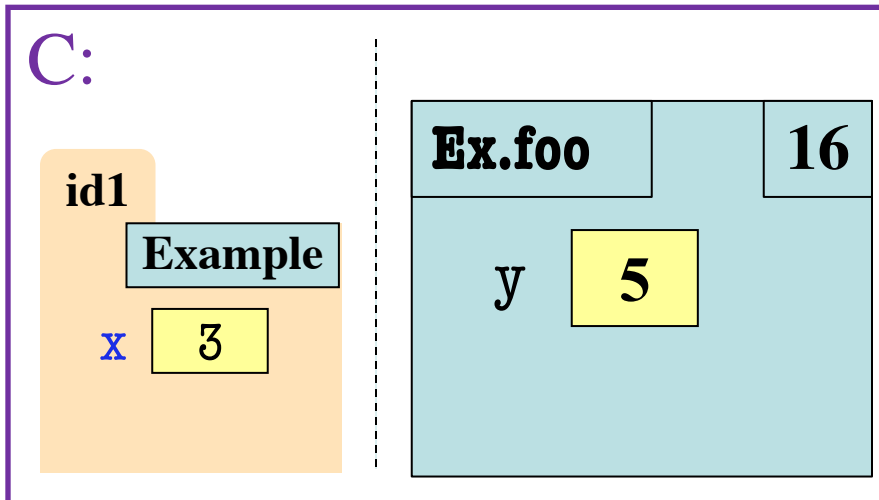
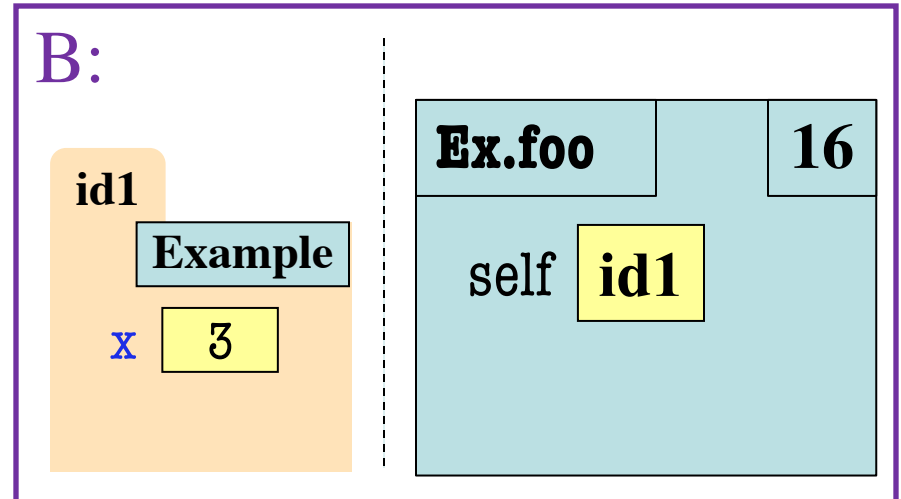
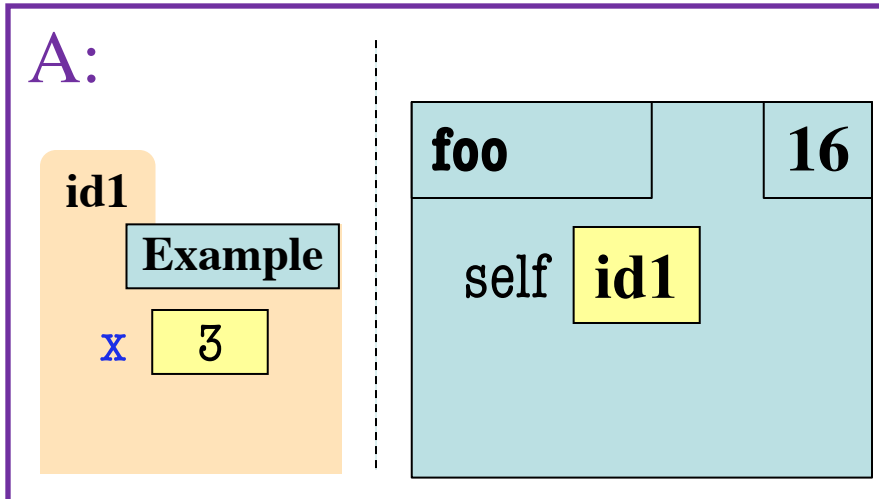
```
class Example(object):  
12  def __init__(self,x):  
13  |   self.x = x  
14  |  
15  def foo(self,y):  
16  |   x = self.bar(y+1)  
17  |   return x  
18  |  
19  def bar(self,y):  
20  |   self.x = y-1  
21  |   return self.x
```

```
>>> 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?



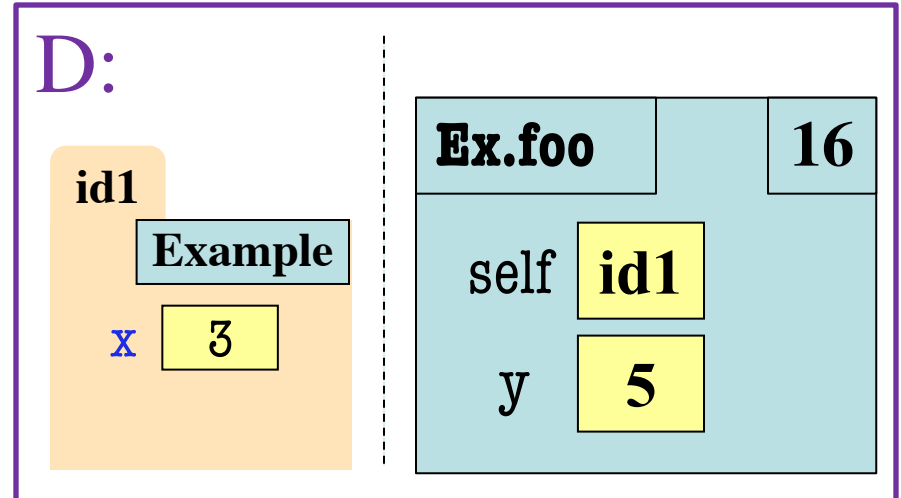


# A Class Definition

```
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:**

id1

Example

x 3

<b>Ex.foo</b>		16
self id1	y	5
	x	5

**B:**

id1

Example

x 3

<b>Ex.foo</b>		16
self id1	y	5
<b>Ex.bar</b>		20
self id1	y	6

**C:**

id1

Example

x 3

<b>Ex.foo</b>		17
self id1	y	5
<b>Ex.bar</b>		20
self id1	y	6

**D:**

id1

Example

x 3

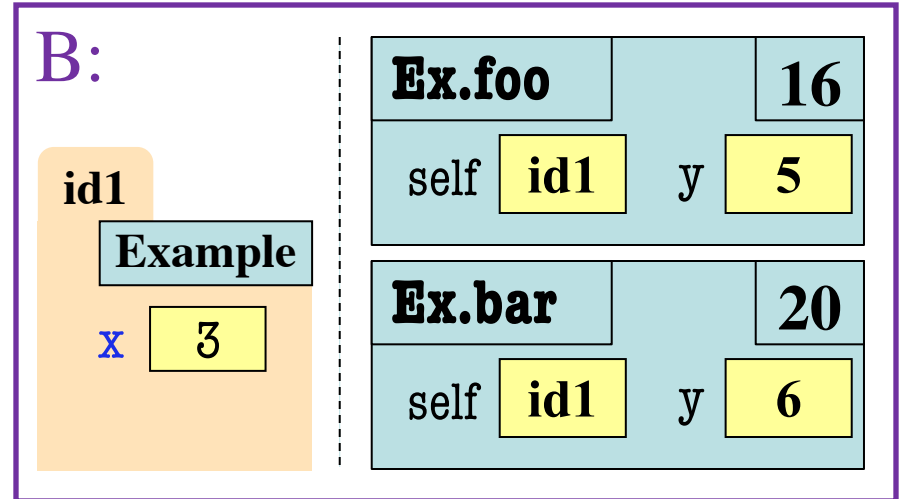
<b>Ex.foo</b>		16
self id1	y	5
<b>Ex.bar</b>		20
	y	6

# 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?

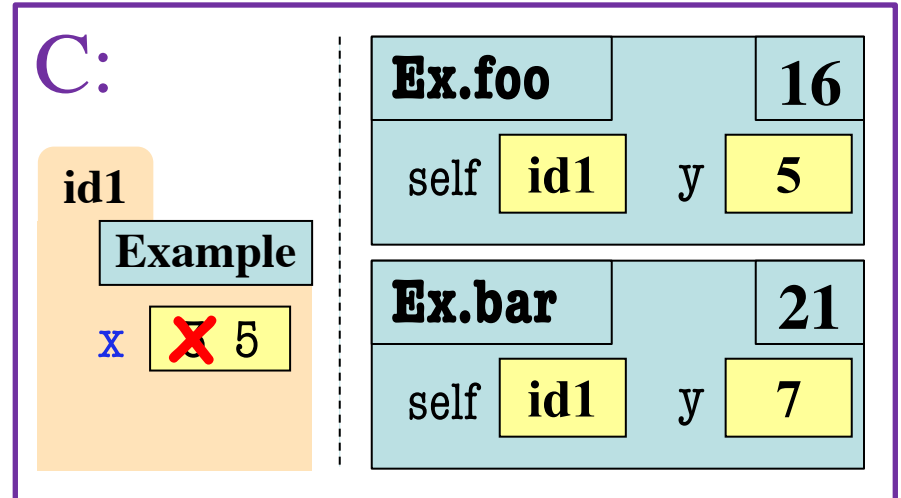
<p><b>A:</b></p> <p>id1</p> <p>Example</p> <p>x <del>5</del></p>	<table border="1"><tr><td><b>Ex.foo</b></td><td></td><td>16</td></tr><tr><td>self</td><td>id1</td><td>y 5</td></tr><tr><td><b>Ex.bar</b></td><td></td><td>21</td></tr><tr><td>self</td><td>id1</td><td>y 6</td></tr><tr><td></td><td></td><td>x 5</td></tr></table>	<b>Ex.foo</b>		16	self	id1	y 5	<b>Ex.bar</b>		21	self	id1	y 6			x 5
<b>Ex.foo</b>		16														
self	id1	y 5														
<b>Ex.bar</b>		21														
self	id1	y 6														
		x 5														
<p><b>C:</b></p> <p>id1</p> <p>Example</p> <p>x <del>5</del></p>	<table border="1"><tr><td><b>Ex.foo</b></td><td></td><td>16</td></tr><tr><td>self</td><td>id1</td><td>y 5</td></tr><tr><td><b>Ex.bar</b></td><td></td><td>21</td></tr><tr><td>self</td><td>id1</td><td>y 6</td></tr></table>	<b>Ex.foo</b>		16	self	id1	y 5	<b>Ex.bar</b>		21	self	id1	y 6			
<b>Ex.foo</b>		16														
self	id1	y 5														
<b>Ex.bar</b>		21														
self	id1	y 6														
<p><b>B:</b></p> <p>id1</p> <p>Example</p> <p>x 3</p>	<table border="1"><tr><td><b>Ex.foo</b></td><td></td><td>16</td></tr><tr><td>self</td><td>id1</td><td>y 5</td></tr><tr><td><b>Ex.bar</b></td><td></td><td>21</td></tr><tr><td>self</td><td>id1</td><td>y 6</td></tr><tr><td></td><td></td><td>x 5</td></tr></table>	<b>Ex.foo</b>		16	self	id1	y 5	<b>Ex.bar</b>		21	self	id1	y 6			x 5
<b>Ex.foo</b>		16														
self	id1	y 5														
<b>Ex.bar</b>		21														
self	id1	y 6														
		x 5														
<p><b>D:</b></p> <p>id1</p> <p>Example</p> <p>x <del>5</del></p>	<table border="1"><tr><td><b>Ex.foo</b></td><td></td><td>16</td></tr><tr><td>self</td><td>id1</td><td>y 5</td></tr><tr><td><b>Ex.bar</b></td><td></td><td>21</td></tr><tr><td>self</td><td>id1</td><td>y 6</td></tr><tr><td><b>RETURN</b></td><td></td><td>5</td></tr></table>	<b>Ex.foo</b>		16	self	id1	y 5	<b>Ex.bar</b>		21	self	id1	y 6	<b>RETURN</b>		5
<b>Ex.foo</b>		16														
self	id1	y 5														
<b>Ex.bar</b>		21														
self	id1	y 6														
<b>RETURN</b>		5														

# 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:**

id1

Example

x 5

<b>Ex.foo</b>		16
self	id1	y 5
<b>Ex.bar</b>		
self	id1	y 6
RETURN	id1	

**B:**

id1

Example

x 5

<b>Ex.foo</b>		16
self	id1	y 5
<b>Ex.bar</b>		
self	id1	y 6
RETURN	5	

**C:**

id1

Example

x 5

<b>Ex.foo</b>		16
self	id1	y 5
<b>Ex.bar</b>		22
self	id1	y 6
RETURN	5	

**D:**

id1

Example

x 5

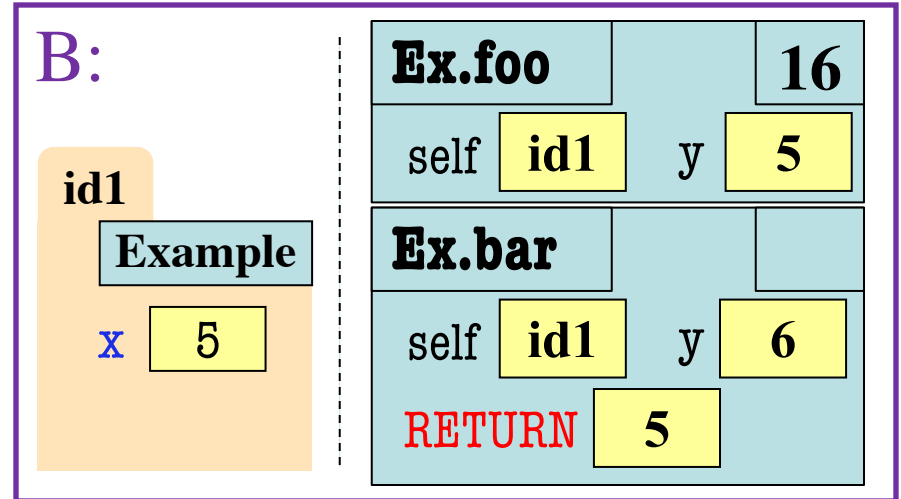
<b>Ex.foo</b>		17
self	id1	y 5
	x	5

# 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):  
13     x = 10  
14  
15     def __init__(self,x):  
16         self.x = x  
17         self.foo(x+1)  
18  
19     def foo(self,y):  
20         self.y = y
```

What is inside  
the **class folder**?



# Which One is Closest to Your Answer?

A:

**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
```

# 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):
```

```
13     x = 10
```

```
14
```

```
15     def __init__(self,x):
```

```
16         | self.x = x
```

```
17         | self.foo(x+1)
```

```
18
```

```
19     def foo(self,y):
```

```
20         | 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-coordinate  
    Precondition: ????  
  
    Parameter y1: The y-coordinate  
    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-coordinate  
    Precondition: ????  
  
    Parameter y1: The y-coordinate  
    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?