Quiz Prep: Environment Diagram Questions

Note

Note that these questions do not affect your grade in 6.009 in any way! They are presented here for two reasons: to provide some extra practice with environment diagrams leading up to the quiz, and to help test the tool used below (for drawing and testing the diagrams), which is being developed as part of an MEng thesis project.

To assess the effectiveness of hints, some questions will not have hints enabled.

Problem 0 - Self Assessment

Before answering the questions below, how confident do you feel about answering diagramming code using classes, such as drawing an environment diagram for the code below?

```python
class Foo:
    def __init__(self):
        self.count = 0

    def add(x, t):
        x.count += t

f = Foo()
y = f.count
y += 4
f.add(5)
```

On a scale of 1 [very not confident] to 5 [very confident], how confident are you about diagramming code with classes?
`--` ▾

Problem 1 - Diagramming Interface

In order to practice drawing environment diagrams, this question tasks you with using a drag-and-drop interface to construct a diagram.

The later questions on this page will ask you to develop diagrams on your own, but this question is designed to help you familiarize yourself with the interface itself.

We also have some problems from the quiz 1 review materials; if you'd like some additional practice, please try the exercises on this page.
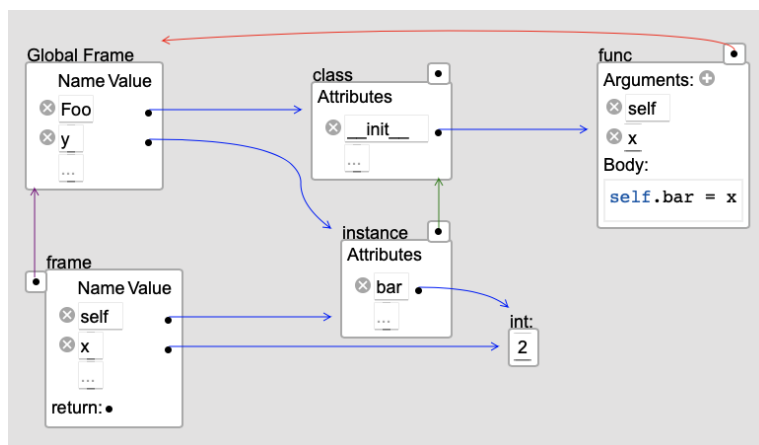
Please use the interface below to re-create the following diagram, representing the state of the program below just before the call to `Foo.__init__` returned.

```python
class Foo():
    def __init__(self, x):
        self.bar = x

y = Foo(2)
```

Note that the global frame has been provided for you.

For instructions on how to use the interface, please click on the **i** icon.

| frame | int | float | str | bool | function | class | instance |

Reset

Global Frame

Name	Value
⊗	...

Problem 2 - Class Definition

Create a diagram representing the result of executing the following code, consisting of a single class definition. Remember that a class object must contain its attributes: the variables and methods defined within the class.
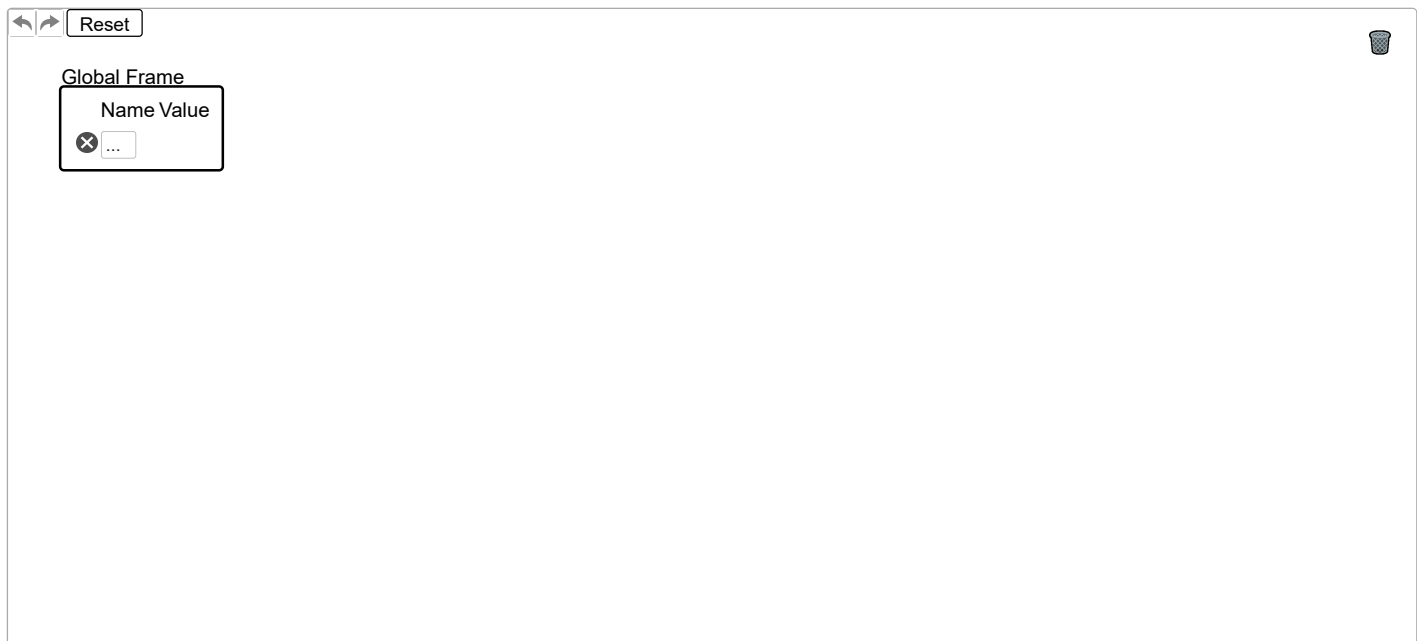
```
class Foo():
    w = 8

    def __init__(self, name):
        self.name = name
```

▶ Show/Hide Hints

| frame | int | float | str | bool | None | function | class | instance |

Problem 3 - Creating Class Instances

If an __init__ function is defined for the class, calls to that custom __init__ should be diagrammed.

Global Frame

Name	Value
⊗ ...	

For this problem, diagram the result of executing the following code. Include each of the frames generated by function calls. For this problem, there should be three total frames: the global frame (already provided), one generated by a call to `Foo.__init__` and one generated by a call to `add`. Parts of the diagram have been filled in for you.

```python
class Foo:
    def __init__(self):
        self.count = 0

    def add(x, t):
        x.count += t

f = Foo()
y = f.count
y += 4
f.add(5)
```

▶ Show/Hide Hints

i	frame	int	float	str	bool	None	tuple	list	function	class	instance

Problem 4 - Class Inheritance

For this problem, diagram the result of executing the following code. Remember to indicate the parent class of subclasses.

```python
class Foo():
    b = 3

class Bar(Foo):
    def increase(self):
        self.b += 4

x = Bar.b
```

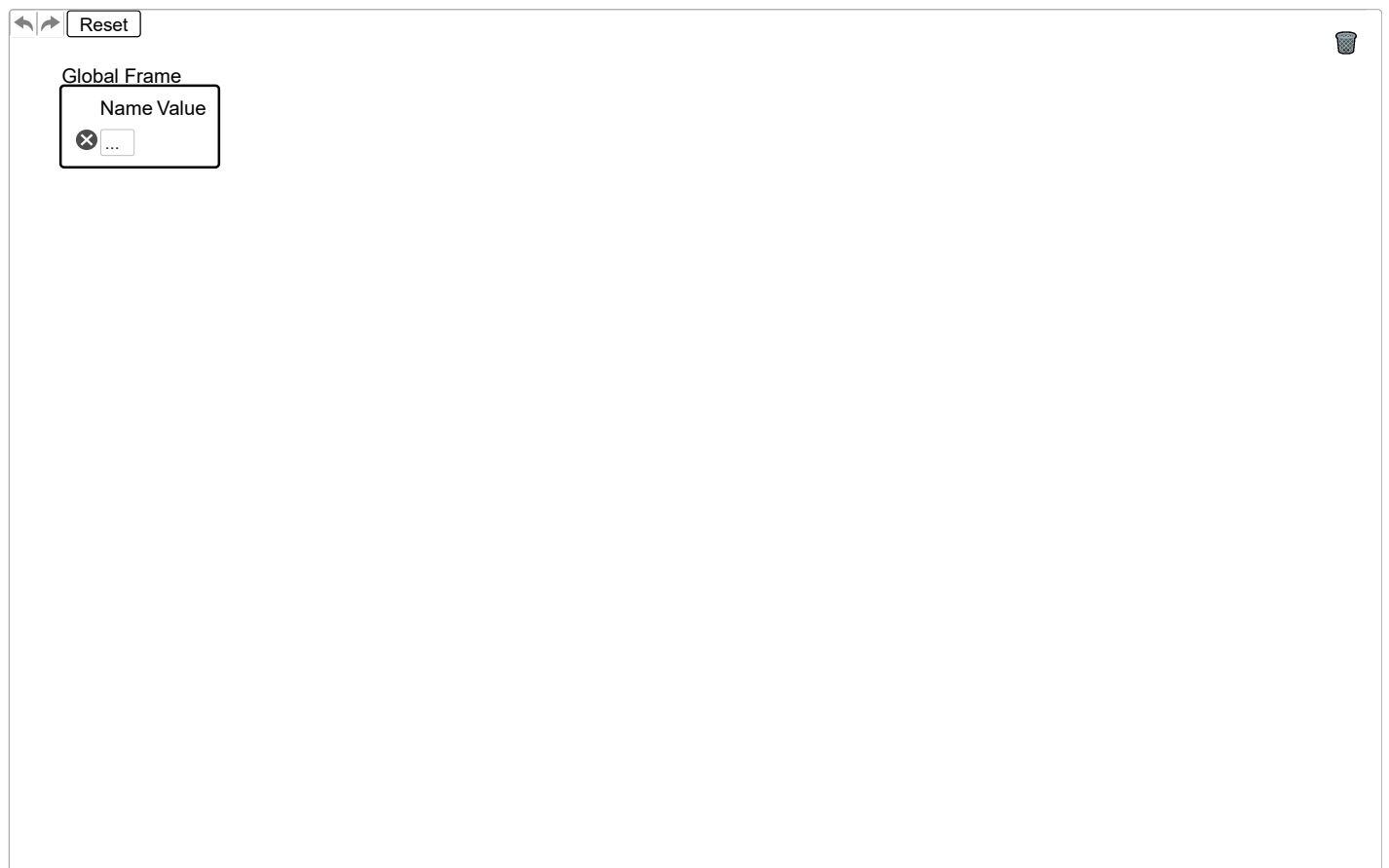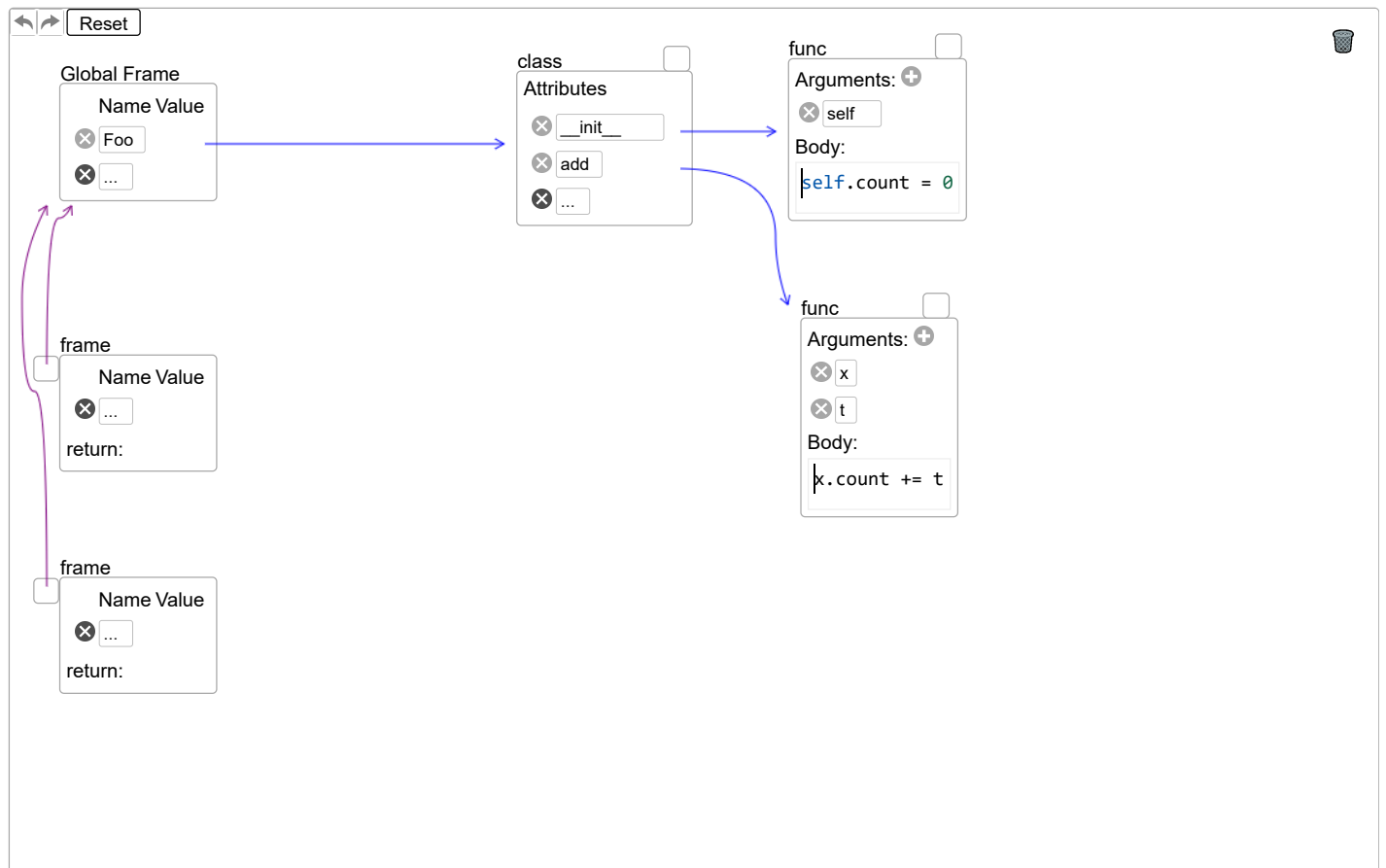| i	frame	int	float	str	bool	None	function	class	instance

Problem 5 - Class Inheritance 2

For this problem, diagram the result of executing the following code. Since there are no `__init__` functions defined for the classes, you should not diagram a frame for the function call of `Bar()`. Some of the diagram has been filled in for you.
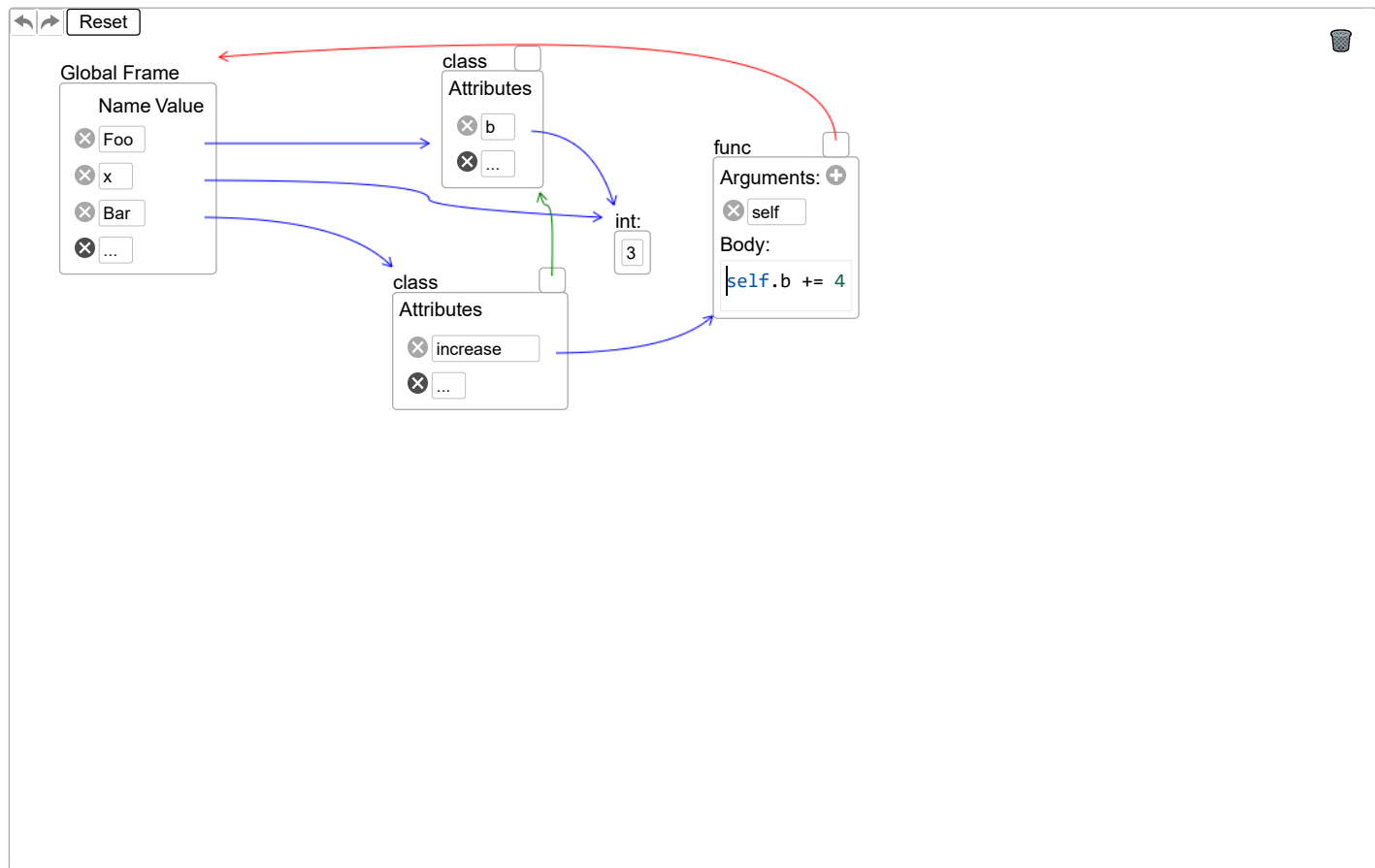
Global Frame

Name	Value
Foo	
...	

class

Attributes
- __init__
- add
- ...

func

Arguments: self

Body:
```
self.count = 0
```

func

Arguments:
- x
- t

Body:
```
x.count += t
```

frame

Name	Value
...	

return:

frame

Name	Value
...	

return:

Global Frame

Name	Value
...	

```python
class Foo():
    b = 3

class Bar(Foo):
    def increase(self):
        self.b += 4

x = Bar.b
```

```
y = Bar()
y.increase()
```

i | frame | int | float | str | bool | None | function | class | instance

Reset

Global Frame

Name Value
⊗ Foo
⊗ x
⊗ Bar
⊗ ...

class ☐
Attributes
⊗ b
⊗ ...

class
Attributes
⊗ increase
⊗ ...

int:
3

func
Arguments: ⊕
⊗ self
Body:
self.b += 4

Problem 6 - Code to Generate a Diagram

Consider the following diagram:
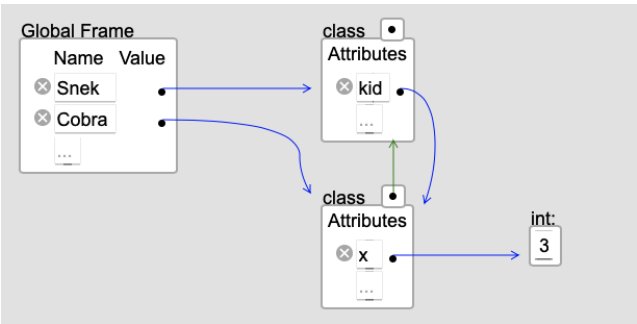
In the box below, enter any valid Python program that, when executed, would lead to the diagram above.

```
1 |
```

Survey

Compared to before this exercise, on a scale of 1 [much less confident] to 5 [much more confident], how has your confidence in diagramming code with classes changed? -- ▾

On a scale of 1 [very difficult] to 5 [very easy], how easy did you find using the interface to depict your diagrams? -- ▾

On a scale of 1 [very unhelpful] to 5 [very helpful], how helpful did you find automated hints while constructing your diagrams? (N/A if not applicable) -- ▾

Please provide any comments or suggestions about the diagraming question tool.