

---

# CS 61A      Structure and Interpretation of Computer Programs

## Spring 2013

---

QUIZ SOLUTIONS

### INSTRUCTIONS

- You have 20 minutes to complete the quiz.
- The quiz is closed book, closed notes, closed computer, and closed calculator.
- Mark your answers in the space provided.

Last name	
First name	
SID	
Login	
TA & section time	

**For staff use only**

Total
/2

1. Suppose the following code is located in `guide.py`:

```
def dont(x):
    return mul(x, x)

def panic(y):
    return mul(x, y)
```

What does the interpreter display when you execute

```
python -i guide.py
```

from the terminal? Assume that `guide.py` is located in the current directory, and ignore any standard startup message. Circle one of the following:

**An error**

**Just a prompt (i.e. >>>)**

**Something else**

**Explanation:** Many students thought there would be an error here because `mul` hasn't been imported yet, or because the return statement of `panic` would fail with no outer `x` defined.

However, the body of a function isn't evaluated when it is defined, so there's no error here.

Now suppose that you enter the following:

```
>>> from operator import mul
>>> dont(3)
```

What does Python display? Circle one:

**An error**

**Just a prompt (i.e. >>>)**

**Something else**

**Explanation:** Here, `mul` is bound in the global frame when `dont` is called, so there's no error. We expect Python to display the return value of the call expression.

Now suppose that you enter the following:

```
>>> panic(3)
```

What does Python display? Circle one:

**An error**

**Just a prompt (i.e. >>>)**

**Something else**

**Explanation:** Here we expect `panic(3)` to fail since there is no `x` for it to reference. The result is an error.

**Question 1 scoring:** +0.5 points for all three parts correct.

2. Consider the evaluation of the following call expression:

```
add(max(3, 4), sub(2, mul(3, 5)))
```

Assume that all names in `operator` have been imported.

In what order are the following functions *applied* by the interpreter? Fill in a number between 1 and 4 in each box.

add	max	sub	mul
4	1	3	2

**Explanation:** A common mistake was mixing up *evaluated* vs. *applied*, which would result in the ordering [1, 2, 3, 4].

Another common mistake was ordering `mul` before `max`. Operands are evaluated from left to right, so `max` must be applied before any operator in the second operand to `add`.

Remember that the operator and operands of a call expression are evaluated *before* the function is applied. Applying a function is the second step in the two-step process of evaluating a call expression - that is, applying a function and evaluating a call expression are *not* the same thing.

**Question 2 scoring:** +0.5 points for the correct ordering.

3. Fill in the environment that results from executing the code below until the entire program is finished. Make sure to add all missing names, values, and frame labels, and show the return value for each frame. *You may not need to use all of the spaces or frames.*

---

```

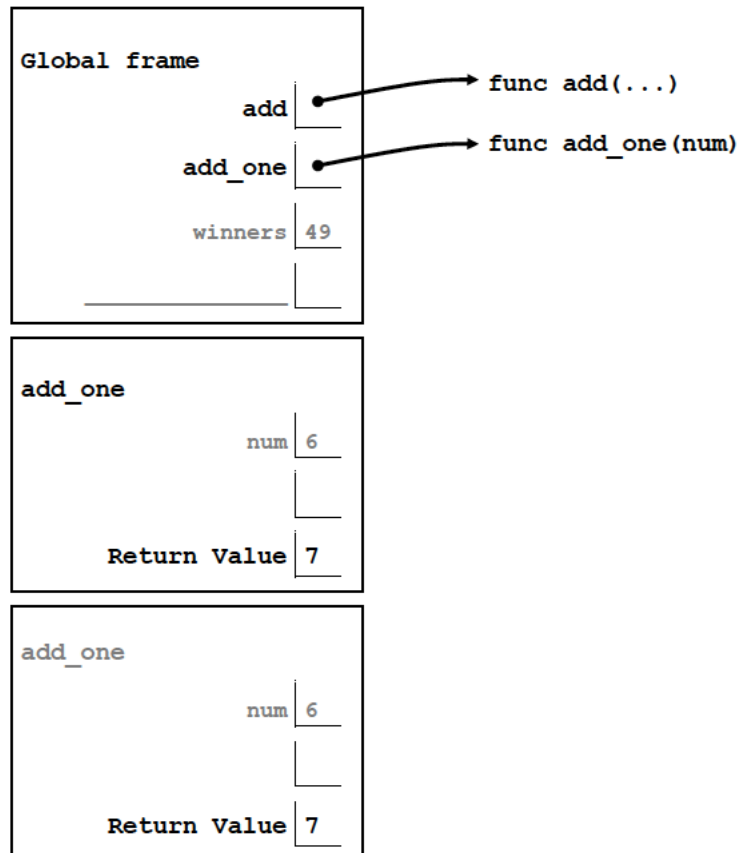
from operator import add

def add_one(num):
    return add(num, 1)

winners = add_one(6) * add_one(6)
print(winners)

```

---



**Explanation:** Many students added unnecessary bindings in their diagrams. Remember that only function definition (i.e. `def...`), assignment (i.e. `x = ...`), function calls, and import statements can add bindings to frames. In the third case, the bindings will only be added to the new local frame.

A common mistake was binding `print` to 49. Only `winners` should be bound to 49, as a result of its assignment. Some students added a binding for `print` to a function value, which is unnecessary but not incorrect, so we didn't take off points in that case.

There are two identical local frames, each resulting from a call to `add_one`. Remember that there should be one local frame for every user-defined function call. Common mistakes include making only one `add_one` frame and making a frame for `winners`. Each local frame should contain solely a binding for `num`, the argument to `add_one`. Another common error was adding a binding for `add`. `add` isn't being redefined or assigned to, so no binding is necessary.

**Question 3 scoring:** +0.5 points for correct global frame, +0.5 points for correct local frames.

4. **(For fun only.)** Name the only two quarterbacks to lead the San Francisco 49ers to a Super Bowl victory.

Steve Young and Joe Montana

Still the case. :( Oh well, there's always next year.