Announcements

**Additional space in labs:**
We have added some space and staffing to the 12:20 and 1:25 labs on Tuesday. There is still space to move into these labs.

**Assignment 1** is out! Grab a printed copy today, and refer to the same text on the website. This assignment is due **February 18**, and you will re-submit until it’s all correct.

Printed copies are coming in a few minutes!
How Do Functions Work?

To evaluate a function call expression:
1. Create a frame for the call
2. Assign arguments to parameters
3. Execute function body
4. Erase the frame

The value of the function call expression is the returned value.

```
def add_em(x, y):
    return x + y

print add_em(3, 4)
```

Return value: 7
How Do Functions Work?

To evaluate a function call expression:
1. Create a frame for the call
2. Assign arguments to parameters
3. Execute function body
4. Erase the frame

The value of the function call expression is the returned value

```
def add_em(x, y):
    1  sum = x + y
    2  return sum
```

```
print add_em(3, 4)
```

Function call

```
add_em: 2
```

Function definition

```
def add_em(x, y):
    1  sum = x + y
    2  return sum
```

Function call

```
add_em: 2
```

Function definition

```
def add_em(x, y):
    1  sum = x + y
    2  return sum
```

Statement numbers

```
def add_em(x, y):
    1  sum = x + y
    2  return sum
```

Function name

```
def add_em(x, y):
    1  sum = x + y
    2  return sum
```

Program counter

```
def add_em(x, y):
    1  sum = x + y
    2  return sum
```

Local variables

```
def add_em(x, y):
    1  sum = x + y
    2  return sum
```

Parameters

```
def add_em(x, y):
    1  sum = x + y
    2  return sum
```

Call frame

```
def add_em(x, y):
    1  sum = x + y
    2  return sum
```

Return value: 7
To evaluate a function call expression:
1. Create a frame for the call
2. Assign arguments to parameters
3. Execute function body
4. Erase the frame

The value of the function call expression is the returned value.

Execute this call on paper. What gets printed out?

A: x, y: 1 2
B: x, y: 2 1
C: x, y: 2 2
D: x, y: 1 1
E: I don’t know
To evaluate a function call expression:
1. Create a frame for the call
2. Assign arguments to parameters
3. Execute function body
4. Erase the frame

The value of the function call expression is the returned value
To evaluate a function call expression:
1. Create a frame for the call
2. Assign arguments to parameters
3. Execute function body
4. Erase the frame

The value of the function call expression is the returned value.
import point
p = point.Point(1,2,3)
q = point.Point(3,4,5)
swap_x(p, q)
print 'p:', p
print 'q:', q

Execute this code on paper.
You will draw 2 objects and a frame.
What is in $p.x$ at the end?

A: 1
B: 2
C: 3
D: I don’t know
```python
def swap_x(p, q):
    t = p.x
    p.x = q.x
    q.x = t

import point
p = point.Point(1,2,3)
q = point.Point(3,4,5)
swap_x(p, q)
print 'p:', p
print 'q:', q
```
function call

c = 2
print g(3)

(9:05 version)

def f(x, y):
    return 3*x + y
c

def g(a):
    b = f(a, c)
    return f(b, a)

function definitions

def f(x, y):
    return 3*x + y

def g(a):
    b = f(a, c)
    return f(b, a)

1. Create a frame for the call
2. Assign arguments to parameters
3. Execute function body
4. Erase the frame
def f(x, y):
    return x * (y**2)

lt_speed = 3e8
print g(3)

def g(m):
    E = f(m, lt_speed)
    return E
How to Draw Things

⟨variable name⟩ ⟨old value⟩ ⟨value⟩

⟨identifier⟩
⟨class name⟩
⟨attributes⟩

⟨class name⟩
⟨methods⟩

⟨function name⟩:⟨program counter⟩
⟨parameters⟩
⟨local variables⟩
Online Python Tutor

pythontutor.com

type in whatever code you want

controls for stepping through code

module (global) variables

frame for call of g

frame for call of f

output from print goes here