CS 150: Functions

Cynthia Taylor
Oberlin College
February 19th 2014

Functions: Motivation

- Let's say you have some code that calculates a student's course mark given the marks on labs, assignments, etc.

- What if you want to apply it to three students?

- We could copy-and-paste the same code three times

Disadvantages of Cut & Paste

- Code length (three times as much code)

- Bugs: if there's a bug or error, we'll replicate it in multiple places

changing code

Analogy

- Imagine you have a cupcake cookbook, with 15 recipes that use buttercream frosting.

- Do you put the buttercream instructions in each recipe? no

reference

Functions

Syntax

```
def <name>():
  <body>
```

– what code we want to run

Syntax

1. Whenever you see <name>(), execute <body>

main()

Calling a function

```
def foo():
  print("Raaarrr I'm a bear")
```

define function

```
foo()
```

calling function

```
def foo():
  print("Raaarrr I'm a bear")

def bar():
  print("Eeek a bear!")

foo()
```

This code will print:

A. Raaarrr I'm a bear

B. Eeek a bear! ← never call bar()

C. Both

D. Neither

E. I don't know

Functions

Syntax

```
def <name>(<parameters>):
  <body>
```

Syntax

1. Whenever you see <name>(<parameters>), execute <body>, using the parameters within the body as appropriate

Parameters

- When we define a function, we specify that it takes zero or more parameters

- When we call a function, we provide a value (an argument) for each parameter; our values are then assigned to the function's parameters

- If there are no parameters, we call the function with empty parentheses

Parameters

```
def f(a, b, c):
    ...

x=5
y=8
z = 10
f(x, y, z)
```

- Parameter passing is just like an assignment statement
- Here, when f starts running, a gets the value of x, b gets the value of y, and c gets the value of z
- Changes to a, b, or c do not change x, y, or z!

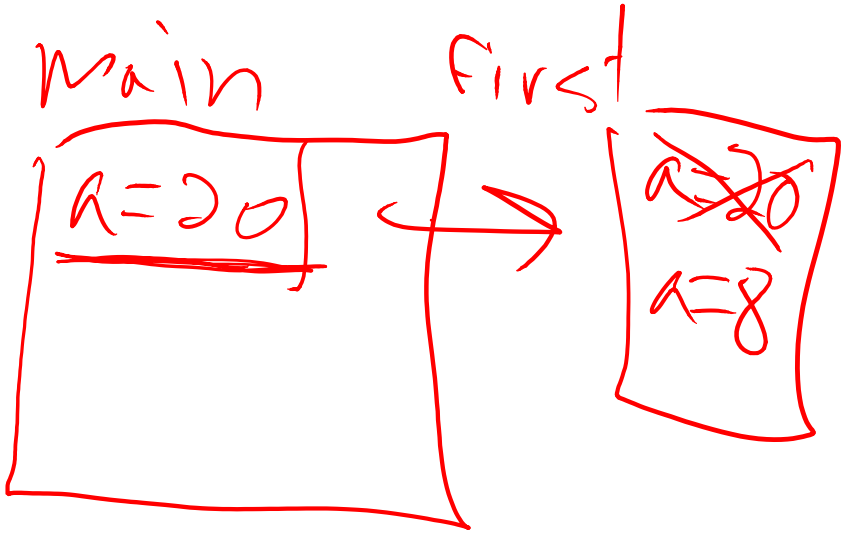
What will the output be?

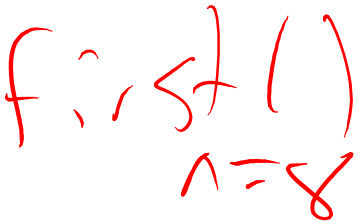
```
def first(a):
    a=8
    return a
a = 20
first(a)
print(a)
```

A. 0

B. 8

C. 20





D. Error, because a cannot be assigned in two places

E. I don't know

return

- If we need a function to produce a value, `return` will exit the function, and replace the function call with the returned value

- If there is no return, the function terminates when it reaches the bottom (and returns None)

What will the output be?

```
def calculate(w, x, y):
    a=x
    b=w+1
    return a + b + 3

print(calculate(3, 2, 0))
```

z = calc(5, 6, 7)

z = 12

```
def foo(x):
    if x<3:
        return True
    return False
```

5

A

9

B

0

C

3

D

return a

~~return b~~

E. I don't know

Which assigns x to 5?

```
def f1():
    return 5

def f2():
    print(5)

def f3():
    return print(5)
```

A. `A.x = f1()`

B. `B.x = f2()` None prints 5

C. `C.x = f3()` ~weirdness

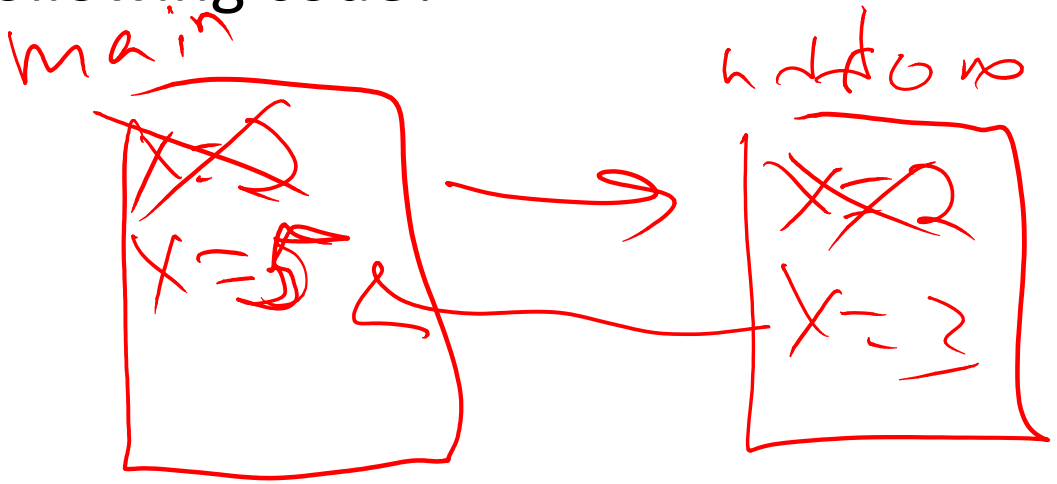
D. All of the above

E. I don't know

What are the bugs in the following code?

```
def add_one(x):
return x + 1

x = 2
x = x + add_one(x)
```



A. No bugs. The code is fine.

B. The function body is not indented.

C. We use x as both a parameter and a variable, but we are not allowed to do that

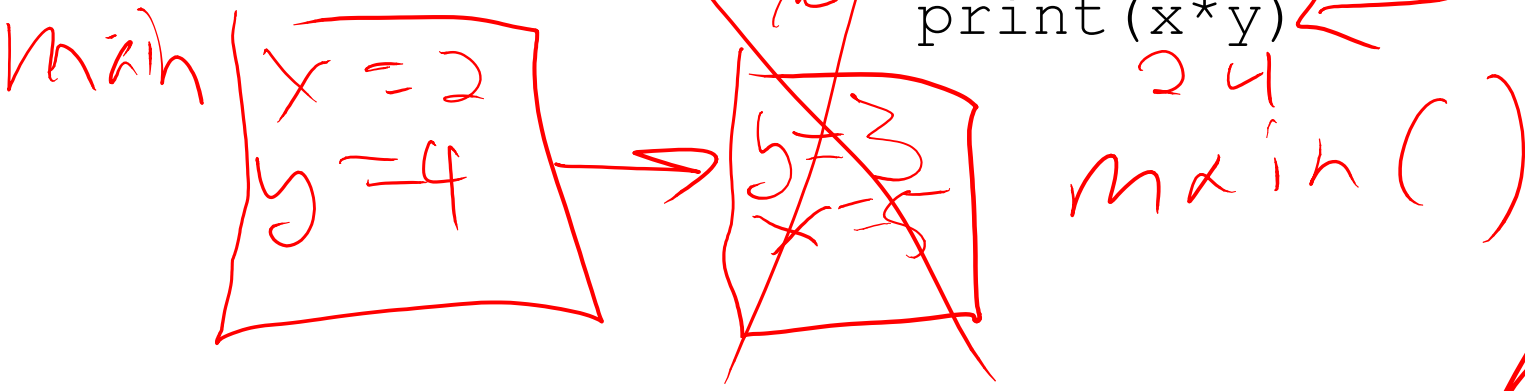
D. B and C

E. I don't know

What will the output be?

```
def odd(y,x):
    y = y +1
    x = x + 1
    print(x*y)

def main():
    x = 2
    y = 4
    odd(x,y)
    print(x*y)
```



A	B	C	D
8 8	15 15	8 15	15 8

E. I don't know

Memory and Functions

```
def odd(y,x):
    y = y +1
    x = x + 1
    print(x*y)
```

```
def main():
    x = 2
    y = 4
    odd(x,y)
    print(x*y)
```

Next Class

- More Functions
- Reading
  - 7.4 – 7.6

- Lab 3 – Due Tuesday at 10pm