Introduction to Programming I in Java (CSE8A)

Winter 2020

Lecture Jan 09, Functions

Upcoming Due Dates

- PA1: Due tonight 10pm (slack in due date until midnight)
- More office/lab hours this afternoon/early evening, ending at 6pm

Getting Help

- Open lab hours today much of the day

- I have office hours today 12:30-2:30 (CSE 3206)
- Alex has office hours 1-3 (CSE B275)

- Keep asking good questions on Canvas! (And check out what your classmates have already asked)

Review: Function Definitions

Assume it takes about $600 to make a $1000 smartphone. Design a function that takes a number of phones sold and returns the **profit** made by making and selling that number of phones.

Which of these is the best choice for the name and arguments of the function?

A. `def profit(cost, revenue):`
B. `def profit(phones_sold):`
C. `def profit(cost, revenue, phones_sold):`
D. `def phones_sold(profit):`
E. All of these have a significant problem

Review: Function Definitions

Assume it takes about $600 to make a $1000 smartphone. Design a function that takes a number of phones sold and returns the **profit** made by making and selling that number of phones.

Which of these is the best definition of the function?

A. `def profit(phones_sold): phones_sold * 400`

B. `profit = phones_sold * 600`

C. `profit = phones_sold * 400`

D. `def profit(phones_sold): return phones_sold * (1000 – 600)`

E. None of these are a good implementation

Review: Function Definitions

Assume it takes about \$600 to make a \$1000 smartphone. Design a function that takes a number of phones sold and returns the **profit** made by making and selling that number of phones.

Which of these are good **tests** for the profit function?

A. `profit(300, 600, 10) # expect answer to be 3000`
B. `profit(2, 3) # expect answer to be 2000`
C. `profit(10) # expect answer to be 4000`
D. None of these are good tests
E. More than one of these are good tests

Comparison Operators and Booleans

New operators

`<  <=  >  >=  ==  !=`

Compare numbers, strings and more for (in)equality. Like other operators, use with nested expressions, variables, etc.

Combining Booleans

New operators

`and` `or` `not`

Take booleans and combine them in various ways.

`b1 and b2` is True when **both** b1, b2 are True

`b1 or b2` is True when **one or both** of b1, b2 are True

Operators and Combining Booleans

Which of these boolean expressions evaluates to True when the variable a is greater than 10?

A. `(a > 10)`
B. `not (a <= 10)`
C. `10 < a`
D. `not (a < 10)`
E. More than one of the above

Operators and Combining Booleans

Which of these boolean expressions evaluates to True when the variable a is greater than 10 and less than 20?

A. `(a > 10) and (a < 20)`
B. `(a < 20) and (a > 10)`
C. `(a >= 20) or (a <= 10)`
D. All of the above
E. A and B only

Functions can return booleans

```
def is_positive(n):
    return n > 0
```

```
>>> is_positive(4)
True
```

$$\begin{tikzcd} \boxed{\texttt{is\_positive(4)}} \arrow[r] & \boxed{\texttt{return 4 > 0}} \arrow[r] & \boxed{\texttt{return True}} \arrow[r] & \boxed{\texttt{True}} \end{tikzcd}$$

Functions can return booleans

Write a function `is_longer_than` that takes a string s and a number n and returns True if the string contains more than n characters.

Before we write the function, what are some **tests** for `is_longer_than`?

Functions can return booleans

Write a function `is_longer_than` that takes a string s and a number n and returns True if the string contains more than n characters.

Now let's try writing the function.

Functions can return booleans

Write a function `between` that takes a string s and a number n and returns True if the string contains more than n characters.

(Practice on your own)

If Statements (or If Blocks)

any number of elif clauses (including 0)

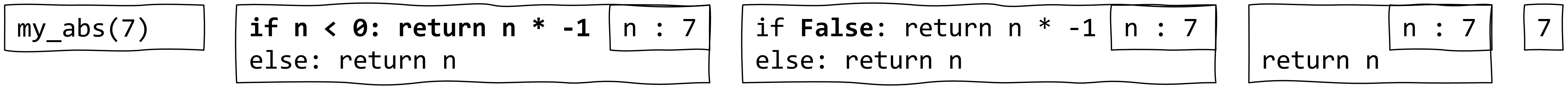
```
if condition1: body1
elif condition2: body2
elif condition3: body3
...
else: body_else
```

optional else clause

Functions that make decisions

```
def my_abs(n):
  if n < 0: return n * -1
  else: return n

>>> my_abs(4)
4
```



```
my_abs(7)
```

```
if n < 0: return n * -1     n : 7
else: return n
```

```
if False: return n * -1     n : 7
else: return n
```

```
                            n : 7
return n
```

```
7
```

Functions that make decisions

Consider letter_grade1-4 on your sheet. Which one of them will produce a return value that **isn't** "C" (it has unexpected behavior) when called with points = 85?

A. letter_grade1(85) is incorrect
B. letter_grade2(85) is incorrect
C. letter_grade3(85) is incorrect
D. letter_grade4(85) is incorrect
E. None of the above (all of them are correct)

Functions that make decisions

Consider letter_grade1-4 on your sheet. Which one of them will produce a return value that **isn't** "C" (it has unexpected behavior) when called with points = 75?

A. letter_grade1(75) is incorrect
B. letter_grade2(75) is incorrect
C. letter_grade3(75) is incorrect
D. letter_grade4(75) is incorrect
E. None of the above (all of them are correct)

Practice with function definitions

Write a function phase_of_water that takes a number representing degrees Celsius and returns "liquid", "solid", or "gas".

What are some good test cases?

Practice with function definitions

Write a function phase_of_water that takes a number representing degrees Celsius and returns "liquid", "solid", or "gas".

Let's try writing the function.