CSE 115

Introduction to Computer Science I
Announcements

Lab activity #1 -> can finish up in second half of lab next week. This is a one-time, not to be repeated deal. Act now :-)  

Lab exam #1 -> you will be expected to write 3 short functions (the first of which is *very* straight forward and worth 40/80 points).
Road map

▶︎ Review ◀

functions and control flow
Review
Control flow
visualizing selection (if)

\[
x = 12 \\
y = 14 \\
\text{if } x < y : \\
\quad z = y \\
a = (x + y) - z
\]
Control flow

visualizing selection (if-else)

\[
\begin{align*}
x &= 12 \\
y &= 14 \\
\text{if } x &< y : \\
\quad z &= y \\
\text{else :} \\
\quad z &= x \\
a &= (x + y) - z
\end{align*}
\]
Control flow

visualizing selection (if-elif-else)

x = 12
y = 14

if x < y:
    z = y
elif x == y:
    z = 0
else:
    z = x

a = (x + y) - z
Road map

Review

functions and control flow
Define a function that takes two values and returns the smaller of the two.

For the purposes of this exercise, do not simply use the min built-in function.

Q: How might we approach this problem?

Brainstorm with your neighbors for a bit, and be prepared to explain (in English, not code) the steps involved.
minimum (of 2)

Here's one possible approach:

```python
def minOf2( x , y ):
```

minimum (of 2)

Here's one possible approach:

```python
def minOf2(x, y):
    if (x < y):
```
minimum (of 2)

Here's one possible approach:

```python
def minOf2( x , y ):
    if ( x < y ):
        return x
```
minimum (of 2)

Here's one possible approach:

```python
def minOf2( x , y ):
    if ( x < y ):
        return x
    else :
        return y
```
minimum (of 2)

Here's one possible approach:

```python
def minOf2(x, y):
    if (x < y):
        return x
    else:
        return y
```

minOf2(4,5) has value 4
minOf2(5,4) has value 4
Define a function that takes three values and returns the smallest of the three.

Q: How might we approach this problem?

Brainstorm with your neighbors for a bit, and be prepared to explain (in English, not code) your approach.
def min0f3( x , y , z ) :
    if ( x < y < z or x < z < y ) :
        return x
    elif ( y < x < z or y < z < x ) :
        return y
    else : return z

min0f3(4,5,6) has value 4
min0f3(5,4,6) has value 4
min0f3(6,5,4) has value 4
minimum (of 3)

another approach

def min0f3ver2( x , y , z ) :
    return min0f2(min0f2(x,y),z)

min0f3ver2(4,5,6) has value 4
min0f3ver2(5,4,6) has value 4
min0f3ver2(6,5,4) has value 4
"homework"
(not to hand in, just to get more practice)

Try the next few problems on your own.
Define a function that computes the arithmetic mean of three values.

Q: How might we approach this problem?

Brainstorm with your neighbors. Be sure to agree on the formula for an arithmetic mean. Be prepared to explain (in English, not code) your approach. If you write code too, that's cool.
arithmetic mean

def a_mean( x, y, z ):
    return ( x + y + z ) / 3
geometric mean

Define a function that computes the geometric mean of three values.

Q: How might we approach this problem?

Brainstorm with your neighbors. Be sure to agree on the formula for an geometric mean. Be prepared to explain (in English, not code) your approach. If you write code too, that's cool.
geometric mean

def g_mean( x, y, z):
    return (x * y * z) ** (1/3)
Define a function that computes either the arithmetic or geometric mean of three values, based on whether the fourth argument to the function is 0 or 1.

0 --> arithmetic mean
1 --> geometric mean

Q: How might we approach this problem?

Brainstorm with your neighbors. Be sure to agree on the formula for an arithmetic mean. Be prepared to explain (in English, not code) your approach. If you write code too, that's cool.
choice of mean

# Comment
# type == 0 --> arithmetic mean
# type == 1 --> geometric mean
# could be extended to more means

def mean( x , y , z , type ) :
    if ( type == 0 ) :
        return a_mean(x,y,z)
    elif ( type == 1 ) :
        return g_mean(x,y,z)

Parameterize the type of mean to compute: no change in program is needed.
mean(a,b,c,t) vs a_mean(a,b,c) g_mean(a,b,c)

Same function is called, but with different arguments.

The name of a function is fixed at runtime.
The values of arguments can vary at runtime.