Function composition

Announcements
● Homework #3 due Thursday

Outline
● Warm-up
● Parameters and variables
● Return values
● Function composition
● Good programming practices

Warm-up
Label each of the following in the program below:
   a) Function header
   b) Loop variable
   c) Function body
   d) Accumulator variable
   e) Docstring
   f) Function definition
   g) Parameter
   h) Function call
   i) Global variable
   j) Return statement
   k) Local variable

import random

NUM_CARDS = 5

def deal_card(suit):
    message = "Dealing a " + suit
    print(message)
    number = random.randrange(2,10)
    return str(number) + " of " + suit + "s"

hand = "Hand: ",
for i in range(0, NUM_CARDS):
    hand = hand + deal_card("heart") + ", ",
print(hand)

Parameters and variables
● Recap: parameters
   ○ Formal --- listed in function definition; specifies what function needs
   ○ Actual --- listed in function call; values passed to function
Variables
  ○ Local variables
    ■ Defined (i.e., first assigned a value) in function
    ■ Only exists in the function in which a value is assigned to it
    ■ Name and value goes away when function finishes => variable is inaccessible outside of the function
  ○ Global variables
    ■ Defined outside of a function
    ■ Can be accessed but not updated inside a function
  ○ Example
    def demoFn():
      varLocal = 10 # Defines local variable called varLocal
      print(varGlobal) # Can access global variable varGlobal within function
      varGlobal = 7 # Defines new local variable called varGlobal

      varGlobal = 5
      demoFn()
      print(varGlobal) # Will still have value 5
      print(varLocal) # Cannot access local variable varLocal outside of function

What is the output of each of the following programs? If the program contains an error, describe and correct the error.

a) def add(a, b):
   """Add two numbers""
   print(a + b)

   def sub(a, b):
     """Subtract two numbers""
     print(a - b)

   add(2, 3)
   x = 8
   y = 2
   sub(x, y)
   add(x + y, 1 + 2)

Output:
5
6
13
b) def multiply():
    """Multiply two numbers""
    print(x * y)

def divide(a, b):
    """Divide two numbers""
    print(a / b)

x = 4
y = 2
multiply()
divide(x, y)
Output:
8
2.0
c) def sum(x, y):
    """Add two numbers""
    print(x + y)

x = 1
y = 2
sum(5, 6)
Output:
11
d) def increment(x):
    """Add one to a number""
    x = x + 1

a = 2
increment(a)
print(a)
Output:
2
e) def ten():
    """Set x to ten""
    x = 10

print(x)
Error --- cannot access variable defined in function

Return values
- Return value --- result provided by the function
  - Not the same as printing out the result!
- How do we make a function return a value? --- add a return statement with the value to return
What does the caller do with the return value?

- Assign it to a variable
- Print it
- Use it as an operand

What is the output of each of the following programs?

a) def magic(a):
   """Manipulate a number""
   return a * 2

   x = 5
   print("x was " + str(x))
   x = magic(x)
   print("now it is " + str(x))

   Output:
   x was 5
   now it is 10

b) def compute_grade(exam1, exam2, exam3):
   """Compute a student's grade""
   sum = exam1 + exam2 + exam3
   return sum / 3

   def letter(grade):
      """Get a student's letter grade""
      if grade >= 90:
         return "A"
      elif grade >= 80:
         return "B"
      elif grade >= 70:
         return "C"
      elif grade >= 60:
         return "D"
      else:
         return "F"

   grade = compute_grade(75, 85, 80)
   print("Grade: " + letter(grade))

   Output:
   Grade: B

Function composition

- Functions can call other functions
  - Function definitions and calls are the same as already discussed
  - Remember: formal parameters and local variables can only be used in the function in which they are defined
What is the output of each of the following programs for each of the provided sequences of input?

a) def contribute(item):
   
   """Ask for person's contributions to something""
   amount = float(input("How much did you contribute to "+item+"? "))
   return amount

   
   def compute_deductions():
      """Calculate total deduction""
      charity = contribute("charity")
      ira = contribute("a traditional IRA")
      itemized = charity + ira
      if itemized < 6350:
         itemized = 6350
      return itemized

   
   def taxable(income, deductions):
      taxable = income - deductions
      return taxable

   
   def main():
      income = float(input("How much did you earn? "))
      deductions = compute_deductions()
      net = taxable(income, deduct)
      print("Your taxable income is $"+str(net))

   
   main()

Inputs A: 99000, 5000, 4000
How much did you earn? 99000
How much did you contribute to charity? 5000
How much did you contribute to a traditional IRA? 4000
Your taxable income is $90000.0

Inputs B: 50000, 500, 1000
How much did you earn? 50000
How much did you contribute to charity? 500
How much did you contribute to a traditional IRA? 1000
Your taxable income is $43650.0
b) def alpha(x, y):
    return x + y

def beta(x):
    z = alpha(x, 3)
    return z**2

i = int(input("Number? "))
print(beta(i))

Input A: 4
49
Input B: 1
16
c) def func1(a, b, c):
    print(b)
    print(a)
    print(c + a)
    return a + b

def func2(a, b, c):
    print(b)
    return c + a

def func3(a, b, c):
    print("W"+b)
    return b + a

a = "X"
b = "Y"
c = "Z"
d = func1(b, c, a)
e = func2(d, a, b)
print(func3(a, e, func1(e, d, c)))
Output:
Z
Y
XY
X
YZ
YZ
YYZ
ZZY
WYYZ
YYZX