CMPT 120
Intro to CS & Programming I

WEEK 12 (Mar. 31-Apr. 4)

— *Jérémie O. Lumbroso* —

Lecture 28:
Order of Execution, variable Scopes

http://www.sfu.ca/~jlumbros/Courses/CMPT120/

Code writing exercise

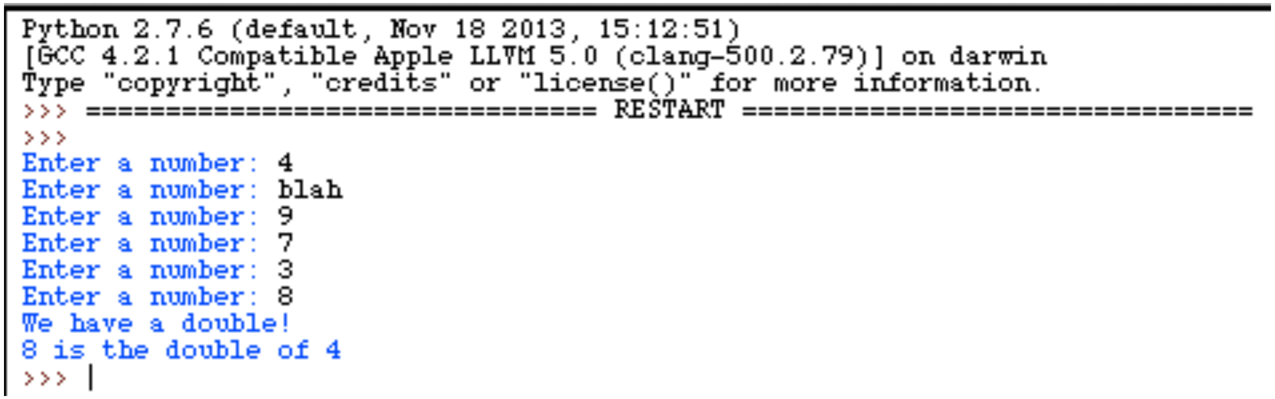
PRACTICE EXERCISE 1

Instructions

- Try to do this in exam condition
  - on paper, and no computer
  - not looking at documents (or only minimally)
- Before trying to write code, **plan**
  - what do you need: lists? for loop(s)? while loop(s)?
  - do you recall similar course example? CodeWrite?
  - imagine this is graded, and do your best, don't just talk about this weekend's gossip (no "Girlfriend, I was like, **so** hungover after KJ's parté on Saturday!!!")

Description

Ask the user to enter one number after another; and stop the process when he enter the double of a number he previously entered.



```
Python 2.7.6 (default, Nov 18 2013, 15:12:51)
[GCC 4.2.1 Compatible Apple LLVM 5.0 (clang-500.2.79)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ================================ RESTART ================================
>>>
Enter a number: 4
Enter a number: blah
Enter a number: 9
Enter a number: 7
Enter a number: 3
Enter a number: 8
We have a double!
8 is the double of 4
>>> |
```

Hints

Ask the user to enter one number after another; and stop the process when he enter the double of a number he previously entered.

What do we need?

- for the input?
- repeating things multiple times
  - do we have a fixed number of iterations? -> for
  - do we iterate when a condition is true/false? -> while
- remembering multiple things
  - do we store numbers in a variable (for instance do we only remember the minimum)?
  - do we need to store them in a list?
- how do we check if the last number is the double of any previous number?
  - iterating over a list?
  - something else?

Solution 1: Test with a `for` loop

```
numbers = []
looping = True

while looping:
  s = raw_input("Enter a number: ")
  if s.isdigit():
    num = int(s)
    numbers.append(num)
    for x in numbers:
      if 2*x == num:
        print "We have a double!"
        print num, "is the double of", x
        looping = False
        break
```

Solution 2: Test Membership with `in`

```
numbers = []

while True:
  s = raw_input("Enter a number: ")
  if s.isdigit():
    num = int(s)
    numbers.append(num)
    if num % 2 == 0 and (num/2) in 
numbers:
      print "We have a double!"
      print num, "is the double of", num/2
      break
```

ORDER OF EXECUTION

Top-level code

- We call "top level" any code that is not in a function (or later a class, or module)
- The "top level" code is generally code that does not have any indentation in front of it

Order of Execution 1

- What is the order of execution of this block of code?

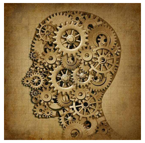
```
def fun(a, b):             #1
  c = a + b*2              #2
  print "inside function"  #3
  return c                 #4

# TOP LEVEL
print "here we start"      #5
val = fun(2, 3)            #6
print val                  #7
```

- Order of execution: 5, 6, 1, 2, 3, 4, 6b, 7
- (Convention 6b means that we go back to that line for assignment)

Pacing and Understanding

How well did you understand today?



- A Too easy, this lecture is way below my abilities
- B Everything went at a good pace, and I am fine
- C Too fast, but I will catch up on my own
- D Too fast, and I need you to slow down
- E I really do not think I can handle this