

## COS 125 Fall 2016 Lab #2 Due Weds 9/14 EoD

NAME Spencer Ward

**Please use a monospaced font such as Courier New, Consolas, or Lucida Console for Python code.** Code and expressions in this document are in Courier New.

Please download this document, type answers or paste code and output into the document, then save or print to a PDF and upload that to complete this lab.

**1. (10 points):** Assume the following:

A = 7

B = 5

C = 9

Determine the truth values (True or False) of the following expressions:

EXPRESSION	VALUE
(A == B)	False
A != 7	False
A != B	True
(A != B) and (C != 11)	True
(A == B) or not (B != C)	False
A - 1 == B + 1	True
not (A != B)	False
A * B - 1 == A * (B - 1)	False
3 ** 3 == A + 2	False
B / 6 == 1	False

**Some hints for printed output in the exercises below:**

- To print more than one value on a line, you can print a comma-separated sequence, e.g.,  
`print 1,2,3`

Will output

`1 2 3`

- If the sequence ends with a comma, Python leaves the line unfinished, so the value printed next appears on the same line.
- A `print` statement all by itself ends the current line and goes to the next line

**2. (15 points):** Write a short statement using a single **for** loop that prints out lines that look like the following:

```
1 1 1 1 1
2 4 8 16 32
3 9 27 81 243
...
25 625 15625 390625 9765625
```

```
for i in range(1, 26):
    print i, i**2, i**3, i**4, i**5
```

**3. (15 points):** Produce the same output at problem 2, this time using two **for** loops, with the second nested inside the first.

```
for i in range(1, 26):
    for n in range(1, 6):
        print i**n,
    print
```

**4. (15 points):** Write a program using two **while** loops that does the same thing as shown in exercise 2 and 3.

```
i = 1
while i < 26:
    n = 1
    while n < 6:
        print i**n,
        n += 1
    print
    i += 1
```



The following two exercises refer to the following program. Please enter it into your computer, save it as a .py file, run it and then do exercises 4 and 5.

```
def print_lyrics():
    print "I'm a lumberjack, and I'm okay."
    print "I sleep all night and I work all day."

def repeat_lyrics():
    print_lyrics()
    print_lyrics()

repeat_lyrics()
```

**5. (15 points) Downey Exercise 3.1:** Move the last line of this program to the top, so the function call appears before the definitions. Run the program and write down the error message that you get. Explain why you get this error message.

```
Traceback (most recent call last):
  File "D:/Users/SWK/Desktop/Google Drive/Documents/2A College Work Freshman/COS
125/Lab 2/lab2-5.py", line 1, in <module>
    repeat_lyrics()
NameError: name 'repeat_lyrics' is not defined
```

Python is interpreted linearly, meaning that when the interpreter reads the function `repeat_lyrics()`, it expects to have a definition loaded in memory that it can refer to. Because the definition for the function is later in the file, the interpreter has not read it yet and has not loaded it into memory. The interpreter does not understand how to execute the line of code, and it crashes.

**6. (15 points) Downey Exercise 3.2:** Move the function call back to the bottom and move the definition of `print_lyrics` after the definition of `repeat_lyrics`. What happens when you run this program? Explain the result.

When the code is arranged this way, the interpreter loads the two functions before running them in the function call `repeat_lyrics()`, meaning that it only needs to load the reference in the first function to the second function. Although this reference is not defined when Python is loading the code, it is by the time the interpreter encounters an execution call.



7. (15 points) Downey Exercise 3.5 (p. 27): Write a function that draws a grid like the following:

```
+ - - - + - - - +
|       |       |
|       |       |
|       |       |
+ - - - + - - - +
|       |       |
|       |       |
|       |       |
+ - - - + - - - +
```

```
def draw_grid():
    gridBar = "+ - - - " * 2 + "+"
    gridBlock = "|       " * 2 + "|"

    for i in range(2):
        print gridBar
        for i in range(3):
            print gridBlock
    print gridBar
```

**Extra Credit (10 points).** Modify the function from problem so that it accepts a parameter called `gridsize` that specifies how many rows and columns to print. In the example above `gridsize` is 2.

```
def draw_grid(gridsize):
    gridBar = "+ - - - " * gridsize + "+"
    gridBlock = "|       " * gridsize + "|"

    for i in range(gridsize):
        print gridBar
        for i in range(3):
            print gridBlock
    print gridBar
```