New York University
School of Continuing and Professional Studies
Division of Programs in Information Technology

Introduction to Python
Exercise Solutions, Session 9

Ex. 9.1 Write a function callme() that prints "function called" every time you call it.

```
def callme():
    print('function called...')

callme()
callme()
callme()
```

Ex. 9.2 Write a function printupper that takes one argument, a string, a prints that string uppercased.

```
def printupper(arg):
    uparg = arg.upper()
    print(uparg)

printupper('hello')
printupper('my you are loud')
printupper('I am not loud.  You are.')
```

Ex. 9.3 Write a function addme that takes two arguments, adds them together and returns the two arguments added / concatenated.

```
def addme(arg1, arg2):
    adx = arg1 + arg2
    return adx

x = addme(4, 5)
print(x)

y = addme('hey', 'you')
print(y)
```

Ex. 9.4 Write a function printlist that takes a list and loops through and prints each element of the list.

```
def printlist(this_list):
    for el in this_list:
        print(el)

printlist([1, 2, 'a', 'b'])
```

Ex. 9.5 Create a module file named yourname.py where yourname is your first name. Do not put a shebang (!#) line at the top.

Create a def hello: function (that prints hello, world!) inside the yourname.py module.

In a file called yourname.py

```
def hello():
    print('hello, world!')
```

Ex. 9.6 Modify the above function to include an optional argument. If name=[something], print hello, [something]! instead of hello, world! But if the name= parameter is not passed, revert to saying hello, world!

```
def hello(name=None):
    if not name:
        name = 'world'

    print('hello, {}!'.format(name))

## even more succinct -- put the default value in the arg list:

def hello(name='world'):

    print('hello, {}!'.format(name))
```

Ex. 9.7 Create a function getlines(filename) that takes a filename, opens the file for that filename, copies the lines of the file (i.e., from readlines()) to a list variable, and then returns the list. In the calling code, call the function with a known filename, and assign the return value of the call to a variable. Loop through the variable (of course it is a list) and print out each line in the file.

```
def getlines(filename):
    fh = open(filename)
    lines = fh.readlines()
    return lines

lines = getlines('../python_data/student_db.txt')

for line in lines:
  print(line)             # prints each line in file
```