

```
##LAB 1 EXERCISES##
```

```
##1. Writing your first Python function
```

```
##Here are some examples of simple functions:
```

```
def f(x):  
    return x**2 + 2  
>>>f(2)  
6
```

```
def ageFunction(name, age):  
    if age >= 20:  
        return "gosh, "+name+" you are getting old!"  
    else:  
        return "you will be "+age+20+" in "+age+20" years!"
```

```
def blastoff(timeInSeconds):  
    while timeInSeconds >= 0:  
        print timeInSeconds  
        timeInSeconds -= 1  
    print "BLAST OFF!"
```

```
'''
```

Write a method `rand_divis_3` that takes no parameters, generates and prints a random number, and finally returns True if the randomly generated number is divisible by 3, and False otherwise. For this method we'll use a new module, the `random` module. At the top of your code, underneath `import math`, add the line `import random`. We'll use this module to generate a random integer using the function `randint`, which works as follows: `random.randint(lo, hi)`

where `lo` and `hi` are integers that tell the code the range in which to generate a random integer (this range is inclusive). 0 to 100 is probably a decent range.

2. Write a method `roll_dice` that takes in 2 parameters - the number of sides of the die, and the number of dice to roll - and generates random roll values for each die rolled. Print out each roll and then return the string, "That's all!"

An example output:

```
>>> roll_dice(6, 3)  
4  
1  
6  
"That's all!"  
'''
```

```
##2. Palindrome?
```

```
in IDLE, set a to a string
```

```
>>> a = "Good morning"
```

```
toy around with []. see what a[0] produces. a[-2]? what does a[3:] produce? a[:3]? a[::-2]?
```

```
##Complete the following codes to test if a string is a palindrome. The function should be able to handle  
##(read: ignore) spaces and capital letters (so 'Never odd or even' should return True). Please refer to Python's  
##online documentation to see some more capabilities strings in Python have. http://docs.python.org/library/string.html
```

```
def is_palindrome(aStr):  
    aStr = #YOUR CODE HERE#           #remove spaces  
    aStr = #YOUR CODE HERE#           #make all lower-case  
    aStrRev = #YOUR CODE HERE#        #create a string that is the reverse of aStr. HINT remember str[]  
    if aStr == aStrRev:  
        return True  
    else:  
        return False
```

```
def is_palindrome2(aStr):  
    aStr = #YOUR CODE HERE#           #recreate aStr in one line so it is lowercase and has no whitespace  
    back_index = -1  
    for char in aStr:  
        if #YOUR CODE HERE#           #conditional if the characters in aStr don't match up forwards and backwards  
            return False  
        back_index -= 1  
    return True
```

##Challenge: Can you recreate this palindrome answer in one line?

##3. Intro to Dictionaries

#Quick Reference D = {} creates an empty dictionary

#D = {key1:value1, ...} creates a non-empty dictionary

#D[key] returns the value thats mapped to by key. (What if there's no such key?)

#D[key] = newvalue maps newvalue to key. Overwrites any previous value. Remember - newvalue can be any valid Python data structure.

#del D[key] deletes the mapping with that key from D.

#len(D) returns the number of entries (mappings) in D.

#x in D, x not in D checks whether the key x is in the dictionary D.

#D.keys() - returns a list of all the keys in the dictionary.

#D.values() - returns a list of all the values in the dictionary.

#For problem 3, write a dictionary that catalogs the classes you took last term - the keys should be the class code (e.g. Phys101), and the values should be the title of the class.

#Then, write a function add class that takes 3 arguments - a class code, class description, and the dictionary

#- that adds new classes to your dictionary. Use this function to add the classes you're taking next term to the dictionary.

#Finally, write a function print classes that takes two arguments - a Course short name (eg 'Phys' or 'Bio')

#and the dictionary you've built - and nicely prints out all the classes you took in that Course.

Example output:

```
>> print_classes('Phys', myClassDict)
```

```
Phys101 - Introductory Mechanics
```

```
Phys102 - Introduction to Electricity and Magnetism
```

```
>> print_classes('Bio', myClassDict)
```

```
No Bio classes taken.
```