CPSC 231 - Lab

FUNCTIONS

Functions in Python

```
def my_functions(<parameter>):
    <body>
    return <value>
```

Calling functions

```
def hello():
    print("Hi there!")
    print("I'm a function!")

print("Good morning")
print("Welcome to class")

hello()

print("And now we're done.")
```

```
sepehr@WNLab-OptiPlex-9020:~/D
Good morning
Welcome to class
Hi there!
I'm a function!
And now we're done.
```

Calling functions inside functions

```
def main():
    print("Hey, I have a massage for you!")
    message()
    print("Goodbye!")

def message():
    print("This is a message!")

#Hey, I have a massage for you!

#This is a message!

#Goodbye!
```

```
global_variable = 1

def increase():
    local_variable = global_variable + 1
    print(local_variable)

def decrease():
    local_variable = global_variable - 1
    print(local_variable)

print(global_variable)
increase()
decrease()
# 2
# 0
# 1
```

Passing Argument to a Function

```
def square(num):
    print(num ** 2)

square(5) # output 25

square()
#TypeError: square() missing 1 required
#positional argument: 'num'
```

Passing multiple Arguments to a Function

```
def average(num1, num2, num3):
    print((num1+num2+num3)/3)

average(1,2,4) # output 2

avarage()
#TypeError: average() missing 3 required
#positional arguments: 'num1', 'num2', and 'num3'
```

```
def change_me(v):
    print ("function got:", v)
    v = 10
    print ("argument is now:", v)
myvar = 5
print ("starting with:", myvar)
change_me(myvar)
print ("ending with:", myvar)
```

```
starting with: 5
function got: 5
argument is now: 10
ending with: 5
```

Returning a value

```
def average(num1, num2, num3):
    return num1+num2+num3)/3

print(average(1,2,4)) # output 2
```

Returning multiple values

```
def find_min_and_max(num1,num2):
    if num1 < num2:
        return num1, num2
    if num2 < num1:
        return num2, num1

min, max = find_min_and_max(3,5)
#min =  3   max = 5
```

eval()

The eval function lets a Python program run Python code within itself.

```
eval("print(\"hi\")")
```

You should import math package if you want to use math functions

```
from math import *

eval("sin(30)"))
```