Week 1

CSE 5246

Due Wednesday, January 20

Welcome to Python 4256. This will be a hands-on class in which you will learn the Python programming language. This is not a class for beginner programmers! It is expected that you are well-versed in at least one other language. To pass the class, you must complete all assignments (but we'll allow you one miss). You should expect the course to take you about three hours per week.

Let's begin.

We'll be working with repl.it, an IDE that allows you to program in the browser. Go to repl.it and sign up for an account as soon as possible. Once you have an account and open a Python window, on the left-side of the screen you'll see the window where we will be typing Python program. The right-side is the command line, where you can also type Python code. We'll start by working with the command line.

Everything is an object in Python. There are no primitive types, as there are in Java. This means, for example, that integers do not have a maximum value. In the example below, the ** operator indicates exponentiation.

```
> x = 2 ** 100
>x 1071508607186267320948425049060001810561404811705533607443750388370351051124936122493198378815695858127594672917553146825187145285692314043598457757469857480393456777482423098542107460506237114187795418215304647498358194126739876755916554394607706291457119647768654216766042983165262438683720566806937 6
```

Python is dynamically typed. While variables don't have types, the objects that they point to do. We can change the type of a variable simply by assigning it to a variable of another type. In the example below, the type() function tells you the type of a variable. str indicates a string.

```
>x=5
> type(x) <class 'int'>
```

```
> x = "foobar" > type(x) <class 'str'>
> x = 3.14
> type(x) <class 'float'>
```

FUNCTIONS
Below is a simple function that adds one to its input and returns the value.

```
def add_one(n):
    return n + 1
```

Functions always begin with def. Next comes the name of the function. It is a convention in Python to use underscores between words in the name of the function, rather than using camelCase as is done in Java. The function header ends with colon.

Unlike other languages, indentation matters in Python to show what code is inside the function. You can use a tab or a few spaces.

The following method computes the hypotenuse of a triangle, given its two legs. The sqrt() function is in the math module, which must be imported.

```
import math
def hypotenuse(x, y):
    return math.sqrt(x ** 2  + y ** 2)
```

COMMENTS
Comments in Python begin with a # sign.

IF-STATEMENTS

if and else work just like in Java:

```
def is_even(n):
    if n % 2 == 0:
        return True
    else:
return False
```

else-if is abbreviated as elif.

```
def sign(n):
    if n > 0:
        return "positive"
    elif n < 0:
        return "negative"
    else:
        return "neither"
```

INCREMENTING
You can increment the value of a variable as follows.

```
x=x+1
```

A shorthand for this is:

```
x += 1
```

LOOPS AND THE RANGE FUNCTION

The range() function returns an object that represents a range of objects. The code below return a range() object that represents the range of integers from 0 to 4.

```
range(5)
```

Typically, range() functions are used in for-loops.

```
> for i in range(5):
> ....print(i)
0
1
2
3
4
```

Note that the range function returns values from 0 up to 1 less than the value passed to the function.

DIVISION

/ is the symbol for standard division.

```
> 3/2
1.5
```

// is integer division.

```
> 3//2
1
```

HOMEWORK

1) Write a function that let's the user play a guessing game. The function should generate a random integer between 1 and 100 and ask the user to enter a number. The function should tell the user if the guess is too low or too high and then allow the user to enter another guess, continuing until the user guesses correctly, at which point the program should print out the total number of guesses used. You can use the following code to generate the random integer:

random.randrange(1, 101)

You'll need to include the random module at the top of your file. Here is a sample run of the program:

```
Enter a guess: 50
To low.
Enter another guess: 75
Too high.
Enter another guess: 63
To low.
Enter another guess: 69
Too high.
Enter another guess: 66
5
```

2) Next write a function that switches the rolls of the computer and user. You the user will pick the random number and the computer will try to guess it. The computer should always guess the correct answer in 7 or fewer guesses.

```
I am guessing 50
Enter L (for low), H (high) or C (correct): L
I am guessing 75
Enter L (for low), H (high) or C (correct): H
I am guessing 62
Enter L (for low), H (high) or C (correct): L
I am guessing 68
Enter L (for low), H (high) or C (correct): H
I am guessing 65
Enter L (for low), H (high) or C (correct): H
I am guessing 63
Enter L (for low), H (high) or C (correct): L
I am guessing 64
Enter L (for low), H (high) or C (correct): C
That is correct.
```

3) Study the article at the following link: https://realpython.com/python-lists-tuples/. Next take the quiz at the bottom of the article. You should take it multiple times until you get every question right. Take a snapshot of the certificate you get at the end of the quiz and paste the certificate into a Word document with your name, the class number, and the homework number (i.e. homework 1) at the top of the page.

WHAT TO SUBMIT

You should submit the code you wrote in repl.it in a file named with your

last name and dot number followed by the homework number. For example, I would submit it with the name "fritz_26_ HW1.py". You can do this be either cutting and pasting your code into a plain text file and saving it with the .py extension, or by clicking on the three dots in the left most window of repl.it, selecting "download as zip", unpacking the zip folder, and finally renaming the file. Next take the python file and the Word document from part (3) and zip them into one folder. Submit the folder on Carmen.