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

Introduction to Python

Homework, Session 1

Please begin with the exercises.

- First, try to complete the exercise from scratch, and if you get stuck, refer to the solution provided in the "Exercise Solutions" document.
- Read carefully and fully understand this solution, then return to the same exercise problem and attempt to again complete the solution from scratch.
- This can be time consuming, but only by fully understanding each solution can you progress meaningfully.
- Exercises are not to be turned in, but feel free to send me questions, especially if you feel you're really not "getting" it -- I can often get you on the right track quickly.

Continue with the homework.

- The homework combines techniques that were used to solve the exercises.
- Each homework is usually accompanied by a "Discussion Document" that goes into detail about the purpose and overall approach of the intended solution.
- The Discussion Document also usually contains an outline in "pseudocode" that can be used to construct the solution, if necessary. A solution should be attempted before using this outline.
- Remember, the course is primarily about internalizing Python's behaviors, not just building programs. If you are concerned about time and deadlines, please don't cut corners -- get in touch with me. I can be flexible about deadlines as long as you are making a steady effort. Better we should have a conversation about the problem than you try to hack something together just to meet the deadline.

EXERCISES

Ex 1.1 Assign two new integer objects and one float object to three variable names. Sum up the three variables and assign the resulting object to a new variable name. Indicate (by printing) the value and type of this resulting object.

Ex 1.2 Assign two new integer objects to two variable names. Multiply the two variables and assign the resulting object to a new variable name. Indicate (by printing) the value and type of this resulting variable's object.

[continued]

Ex 1.3 Starting with the following code (make sure it is included exactly as written) sum up the values to produce the integer value **15**. (Hint: apply a conversion function to each string so it can be used as a number.)

Starter code:

```
var = '5'
var2 = '10'
```

Expected output:

```
15
```

Ex 1.4 Take user input for an integer and print that value doubled.

Sample output (sample user input in bold):

```
Please enter an integer:  5
5 doubled is 10.
```

Ex 1.5 Take user input for a 'place' and then greet the place enthusiastically.

2 Sample program runs (sample user input in bold):

```
Please enter a place name:  Hawaii
Hello, Hawaii!

Please enter a place name:  Katmandu
Hello, Katmandu!
```

Ex 1.6 Take user input for an integer and apply that many exclamation points to the phrase, "Hello, world!" (Hint: use the "string repetition" operator)

2 Sample program runs (sample user input in bold):

```
Please enter an integer:  2
Hello, World!!

Please enter an integer:  11
Hello, World!!!!!!!!!!!
```

Ex 1.7 Assign the float value **35.30** to a variable, then round the value to **35**.

Expected output:

```
35
```

Ex 1.8 Assign the float value **35.359958** to a variable, then round the value to two decimal places.

Expected output:

```
35.36
```

Ex 1.9 Starting with the following code (make sure it is included *exactly as written*), divide the first number by the second number.

```
var = "5"
var2 = "4"
```

expected output:

```
1.25
```

HOMEWORK

Feel free to refer to the discussion document for further detail on how to proceed (including a code outline).

1.1 Exponientation with tidy border: start with the following code, which calls **input()** twice to accept two inputs from the user:

```
var_1 = input('Please enter an integer:  ')
var_2 = input('Please enter another integer:  ')
```

Raise the first number to the power of the second number (you will have to convert them to numbers to do the math). Take the resulting number and determine its string length with **len()** (you will have to convert it to a string to get its string length). Print a line with a ======== border exactly the length of the resulting number (use the * operator to repeat the string '='). On the next line, print the number. Then on a third line print a ======== border the same length as the first.

Sample runs (bold indicates user's input from the keybaord):

```
$ python homework_2.py
please enter an integer:  3
please enter another integer:  14
=======
4782969
=======

$ python homework_2.py
please enter an integer:  3
please enter another integer:  5
===
243
===
```

1.2 Tip calculator: write a restaurant bill calculator that takes three inputs: a restaurant bill total charge, the number of people who will be splitting the bill, and the desired tip in percent of the total bill.

Expected Output (with sample input in bold):

```
Please enter the total bill amount:  120
Please enter the number in your party:  5
Please enter the desired tip percentage (for example, "20" for 20%):  15
A 15.0% tip ($18.0) was added to the bill, for a total of $138.0
With 5 in your party, each person must pay $27.6
```

SUPPLEMENTARY (EXTRA CREDIT) EXERCISES

These assignments are intended to give you additional practice if you have the time and inclination. I will look at any submissions from this section, although I may not have as much time to offer detailed inline comments as I do with the required assignments.

1.3 Price Per Unit Comparison: write a program that takes four inputs: the price of a product, the unit size of that product (for example, "5" for five ounces, or "2.5" for 2.5 litres), the price of a second product, and the unit size of that product (using the same units). The program will then calculate the price per unit of each product and show which product is the better value on that basis.

Program runs (with sample input):

```
please enter the unit size of Product 1:  100
please enter the price of Product 1:  10
please enter the unit size of Product 2:  50
please enter the price of Product 2:  7.50
Product 1 costs $0.1 per unit
Product 2 costs $0.15 per unit
Product 1 is 66.67% the per-unit cost of Product 2
```

1.4 Share Calculator: this program is just a practical demonstration of how easily Python can pull in data from the web for use in your programs. It will only work if your computer is connected to the internet (and there are no firewall or other systems in place that might prevent a network connection from your computer).

Calculate the number of shares of a given stock a given amount of money can buy. Take user input for a stock symbol (for example, "AAPL" (Apple Inc.), "GOOG" (Google Inc.), "DOW" (Dow Chemical Co.) and a cash dollar amount, and calculate how many shares could be bought for the cash.

The program uses a pre-written function (**get_ticker()**, supplied below) to retrieve the stock's current price from the internet to use in the calculation.

[continued]

Starting code:

```
def get_price(symbol):
    import urllib.request, urllib.parse, urllib.error
    import json

    base_url = "http://finance.google.com/finance/info?client=ig&q={}:{}"
    try:
        content = urllib.request.urlopen(base_url.format('NASDAQ',
symbol)).read()
    except IOError:
        exit('Error:  cannot connect')

    content_json = content.decode('utf-8')

    try:
        obj = json.loads(content_json[3:])
    except ValueError:
        exit('Error:  stock symbol ' + symbol + ' not found')

    current_price = obj[0]['l'].replace(',', '')

    return float(current_price)

aa = input('please input the stock symbol you would like to buy:  ')
bb = input('please input the cash you have to invest:  ')

cc = get_price(aa)      # returns a float price

## your code goes here - calculate the # of shares
## that can be bought with the user's cash
```

Expected output (with sample input and current stock price):

```
please input the stock symbol you would like to buy:  UNP
please input the cash you have to invest:  10000
with $10000 you can buy 90 shares of UNP at $110.62/share
```