Lab Questions: Lab Session 4

Deadline: 13.09.2017 11:59pm SGT

Complete all assignments below. For those questions that are marked with an asterisk *, i.e. questions 6 and 10, create the script files as requested. Once you are done with it, submit the file via `iNTU`.

Important!!! Make sure your scripts work properly, as we give 0 marks otherwise. Please name the scripts according to the requirements, and upload each file separately and not in a Zip file or similar. The submission system closes at the deadline. Hence after that, you will get no marks for your solution.

1. Write a simple script called `spherevolume.py` that will calculate the volume of a sphere by the formula

$$V = \frac{4\pi}{3} r^3$$

where r is the radius of the sphere. In the script, assign a value to a variable `r` for the radius. Then, using this variable, compute the volume and store it in `vol`. Include comments in the script to clearly explain what you are doing.

Solution:

spherevolume.py

```
# This script calculates the volume of a sphere

import math

# Ask the user for the value of the radius
print('Calculation of the volume of sphere')
r = float(input('Please enter the radius: '))

# Calculate the volume
vol = ( 4 * math.pi) / 3 * r**3

# Output the result
print('Volume of sphere with radius %.2f is %.2f' % (r, vol))
```

2. The atomic weight is the weight of a mole of atoms of a chemical element. For example, the atomic weight of oxygen is 15.9994 and the atomic weight of hydrogen is 1.0079. Write a script that will calculate and print the molecular weight of hydrogen peroxide, which consists of two atoms of hydrogen and two atoms of oxygen. It is up to you how to name the input and the output variables. Include comments in the script.

Solution:

molweight.py

```
# Calculates the molecular weight of hydrogen peroxide
```

```
# Initialize the atomic weights for oxygen and hydrogen
atWtOxygen = 15.9994
atWtHydrogen = 1.0079

# Hydrogen peroxide is 2 atoms of hydrogen and 2 of oxygen
molWtHydrogenPeroxide = 2*atWtHydrogen + 2*atWtOxygen
print(molWtHydrogenPeroxide)
```

3. Write an input statement that will prompt the user for the name of a chemical element as a string. Then, print the length of the string.

   Solution:
   ```
   >>> elemname = input('Enter a chemical element:  ')
   >>> print(len(elemname))
   ```

4. Test your knowledge and understanding of the `print` function for integers (give both the old-style and new style formatting answers). Use `print` function to print the integer 12345,

   Solution:
   ```
   intnum = 12345
   ```

   (a) without specifying any field width

   Solution:
   ```
   >>> print('The number is %d' % (intnum))
   The number is 12345
   >>> print('The number is {}'.format(intnum))
   The number is 12345
   ```

   (b) in a field width of 5

   Solution:
   ```
   >>> print('The number is %5d' % (intnum))
   The number is 12345
   >>> print('The number is {:5d}'.format(intnum))
   The number is 12345
   ```

   (c) in a field width of 8

   Solution:
   ```
   >>> print('The number is %8d' % (intnum))
   The number is    12345
   >>> print('The number is {:8d}'.format(intnum))
   The number is    12345
   ```

   (d) in a field width of 3

   Solution:
   ```
   >>> print('The number is %3d' % (intnum))
   ```

```
The number is 12345
>>> print('The number is {:3d}'.format(intnum))
The number is 12345
```

5. Test your knowledge and understanding of the `print` function for real numbers or float (give both the old-style and new style formatting answers). Use `print` function to print the real number 12345.6789,

   Solution:
   ```
   realnum = 12345.6789
   ```

   (a) without specifying any field width

   Solution:
   ```
   >>> print('The number is %f' % (realnum))
   The number is 12345.678900
   >>> print('The number is {}'.format(realnum))
   The number is 12345.6789
   ```

   (b) with 7 decimal places

   Solution:
   ```
   >>> print('The number is %.7f' % (realnum))
   The number is 12345.6789000
   >>> print('The number is {:.7f}'.format(realnum))
   The number is 12345.6789000
   ```

   (c) in a field width of 10 with 4 decimal places

   Solution:
   ```
   >>> print('The number is %10.4f' % (realnum))
   The number is 12345.6789
   >>> print('The number is {:10.4f}'.format(realnum))
   The number is 12345.6789
   ```

   (d) in a field width of 10 with 2 decimal places

   Solution:
   ```
   >>> print('The number is %10.2f' % (realnum))
   The number is   12345.68
   >>> print('The number is {:10.2f}'.format(realnum))
   The number is   12345.68
   ```

   (e) in a field width of 6 with 4 decimal places

   Solution:
   ```
   >>> print('The number is %6.4f' % (realnum))
   ```

```
The number is 12345.6789
>>> print('The number is {:6.4f}'.format(realnum))
The number is 12345.6789
```

(f) in a field width of 2 with 4 decimal places

Solution:

```
>>> print('The number is %2.4f' % (realnum))
The number is 12345.6789
>>> print('The number is {:2.4f}'.format(realnum))
The number is 12345.6789
```

6. * Write a script <YourMatricNo>_Lab4_CalcAreaTrap.py to prompt the user for the lengths of the parallel sides, and the height of a trapezoid, and print its area with 2 decimal places. Put comments in the script.

Solution:

CalcAreaTrap.py

```
# Calculate the area of a trapezoid

# Prompt the user for the length and width
a = float(input('Enter the length of the first side: '))
b = float(input('Enter the length of the second side: '))
h = float(input('Enter the height: '))

# Calculate and print the area
trap_area = (a+b)/2 * h
print('The area of the trapezoid is %.2f' % (trap_area))
```

7. In the metric system, fluid flow is measured in cubic meters per second (m^3/s). A cubic foot per second (ft^3/s) is equivalent to $0.028 m^3/s$. Write a script titled `flowrate.py` that will prompt the user for flow in cubic meters per second and will print the equivalent flow rate in cubic feet per second. Here is an example of running the script. Your script must produce output in exactly the same format as this:

```
>>> flowrate
Enter the flow in m ^ 3 / s:  15.2
A flow rate of 15.200 meters per sec is equivalent to 542.857 feet per sec
```

Solution:

flowrate.py

```
# Converts a flow rate from cubic meters per second
# to cubic feet per second

cubMperSec = float(input('Enter the flow in m ^ 3 / s:  '))
cubFperSec = cubMperSec / 0.028
print('A flow rate of %.3f meters per sec ' % (cubMperSec) + 'is equivalent to %.3f
    feet per sec\n' % (cubFperSec))
```

```
# To print, we could have instead written:
print('A flow rate of %.3f meters per sec ' % (cubMperSec),end='')
print('is equivalent to %.3f feet per sec\n' % (cubFperSec))
```

8. On average, Singaporeans spend 8% to 10% of their income on food. Write a script that will prompt the user for an annual income. It will then print the range that would typically be spent on food annually. Also, print a monthly range. Make sure your answer is meaningful, i.e. print not only the result but also some text.

   Solution:

   food.py

```
# Calculates and prints the likely $ amount spent on food
# based on annual income

# get the income from the user
income = float(input('Enter your annual income: '))

# print the annual range
print('You are likely to spend between $%.2f' % (.08*income) + 'and $%.2f annually on
    food.' % (.1*income))

# print the monthly range
print('You are likely to spend between $%.2f' % (.08*income/12) + 'and $%.2f monthly
   on food.' % (.1*income/12))
```

9. Write a script that first asks the user to enter a string. Then, the script asks the user to enter two letter positions and eventually prints the new string with the two chosen letters swapped. Here is an example of one execution:

```
Please enter a string: I am studying in NTU
Please enter the index of the first letter (counting starts at 0): 2
Please enter the index of the second letter (counting starts at 0): 7
I um stadying in NTU
```

   Write that script with two different strategies:

   (a) (script 1) by storing the user input in a string

   Solution:

   SwapLetter.py

```
# Asks the user to enter a string and let him choose two letters
# that he would like to swap. Finally prints the new string

# Asks the user to enter a string
my_str = input('Please enter a string: ')

# Prompt the user to choose two letters
```

```
first = int(input('Please enter the index of the first letter: '))
second = int(input('Please enter the index of the second letter: '))

# Compute the new string
# you can't swap the letters directly as strings are immutable
my_str = my_str[:first] + my_str[second] + my_str[first+1:second] + my_str[first]
     + my_str[second+1:]

# Display the new string
print(my_str)
```

(b) (script 2) by storing the user input in a list (you can use the `join` method from string objects)

Solution:

SwapLetter2.py

```
# Asks the user to enter a string and let him choose two letters
# that he would like to swap. Finally prints the new string

# Asks the user to enter a string and type cast it to a list
my_list = list(input('Please enter a string: '))

# Prompt the user to choose two letters
first = int(input('Please enter the index of the first letter: '))
second = int(input('Please enter the index of the second letter: '))

# Compute the new string
# you can swap the letters directly as list are mutable
letter=my_list[first]
my_list[first]=my_list[second]
my_list[second]=letter

# Display the new string by merging the caracters with the join method
print(''.join(my_list))
```

10. * Write a script <YourMatricNo>_Lab4_RandMatrix.py that accomplishes the following task. First, it creates a matrix of size 3×4 of random integers in the range $[10, 99]$. Then it prints the matrix (free format). Finally, it asks the user to enter a row value and a column value, and it prints (in field of width 3) the sum of all elements of the entered row, and the sum of all elements of the entered column.
Hint: use the function `randint` from the numpy random (`https://docs.scipy.org/doc/numpy/reference/generated/numpy.random.randint.html`)
Here is an example of one execution:

```
The generated matrix is:
   [[72    13    78    54]
    [38    49    81    50]
    [95    44    26    68]]

Please enter the row (counting starts at 0): 1
Please enter the column (counting starts at 0): 3
The sum of all elements of row 1 is 218
The sum of all elements of column 3 is 172
```

Solution:

RandMatrix.py

```
# Computes the sum of elements of user chosen row and column of
# a random 3x4 matrix

# import the numpy module
import numpy as np

# Generate the matrix with integers in [10,99]
mat=np.random.randint(low=10,high=100,size=(3,4))

# Display the matrix
print('The generated matrix is:')
print(mat)

# Prompt the user to enter the row and the column numbers
row = int(input('Please enter the row (counting starts at 0): '))
col = int(input('Please enter the column (counting starts at 0): '))

# Compute the sums of elements of the chosen row and column
sum_row = mat[row,0] + mat[row,1] + mat[row,2] + mat[row,3]
sum_col = mat[0,col] + mat[1,col] + mat[2,col]

# Display the sums in a proper format
print('The sum of all elements of row %d is %3d' % (row, sum_row))
print('The sum of all elements of column %d is %3d' % (col, sum_col))
```