Geo1000 – Python Programming – Assignment 0

Due: Thursday, September 3, 2020

Introduction

This handout is to get you started and make your first steps of programming with Python. Helping each other is fine. Try to make sure that your implementation (i.e. the program you make) is your *own*.

This is an ungraded assignment that will not count for your final grade. The results do not need to be handed in.

However, we ask you to check your results with one of your fellow students during the first lab session and to discuss the results you obtained. Did you get the same answers? Did you do the same things?

You should strive to have a working environment on your own laptop *before* the lab session on Thursday, so you can start working on Assignment 1.

1 Installing Python and Using the Interactive Console

You are expected to install Python on your own laptop and make some first steps with the language.

1.1 Installing Python

For the Python course we will be using the official Python *3.x* distribution from the Python website: http://python.org/.

At the Python website you can also find a myriad of information about Python and the official documentation about the language (https://docs.python.org/3/index.html). To get an in-depth and rather formal explanation of the Python programming language, you can take a look at the Python Language Reference (https://docs.python.org/3/reference/index.html). A more informal introduction can be found in the Python tutorial (https://docs.python.org/3/tutorial/index.html).

Download the current 3.x release (at the moment of writing, the version is 3.8.5) and install the **correct** version for your platform (Windows, Mac, Linux) and architecture (most likely *64 bits*). Please follow the video to install and test Python: https://youtu.be/rVb1TqqbPj0. If you're faced with problems during installation, you can get help during the lab session.

1.2 Exercising with Python's interactive console

Start Python's interactive console to perform the following exercises. You can find how to open the console in the video from time 3:32 to time 4:15 of https://youtu.be/rVb1TqqbPj0, which you have already watched during installing Python.

1. Simple Commands.

   Type one of the following one-line commands into Python's interactive console each time, and then press "Enter" to see the output:

   - `credits`

   - `copyright`

   - `license()`

   - `help()`

2. Simple Expressions.

   Type one-line commands to IDLE to compute the following:

   - `12345 + 23456`

   - `98765 - 12345`

   - `128 * 256`

   - `555 / 111`

   - `(18 - 32) * 5 / 9`

   - `-10 * 9 / 5 + 30`

   - `22 / 7`

   - `22.0 / 7.0`

   Did these expressions all give a result you had expected? Use a `print` statement to show your answer in the interactive console.

3. (a) Create an expression that uses the sum operator (+) to sum the individual digits of your student number (as integers).

   (b) Create an expression that multiplies all the digits of your student number (as integers).

   (c) Create a variable `name` to which you assign your name as string.

   (d) Create a variable age to which you assign your age as floating point number.

4. Convert between degrees Celsius and degrees Fahrenheit.

   Convert temperatures from one system to another. The following two formulae convert between $°C$ (Celsius) and $°F$ (Fahrenheit).

   - $°C \times \frac{9}{5} + 32 =° F$

   - $(°F - 32) \times \frac{5}{9} =° C$

   Conversion constants: $32°F = 0°C, 212°F = 100°C$.

   Use the two variables *celsius* and *fahrenheit* to assign the values for the conversion.

   (a) Create a statement to convert 18 °C to °F.

   (b) Create a statement to convert -4 °F to °C.

2 Installing Visual Studio Code

Using Python's interactive console, one can write some simple source code. However, if we want to write a program with complex source code, then we need to save and coordinate the source code in different files with "py" as file extension (e.g., "test.py"). Visual Studio Code (VS Code) is a powerful tool to help us organize source-code files when we are coding. Moreover, VS Code provides rich features to enable us to write source code more efficiently.

Please download VS Code at https://code.visualstudio.com/. Please install VS Code and launch it. In VS Code, please install extension 'Python' so that you can run Python in VS Code. You can find an instruction of installation in the video at https://youtu.be/-nh9rCzPJ20, where you should watch the first ten minutes at least.

3 Using Python in script mode: Wind chill factor – windchill.py

You are expected to create a program as a Python script (i.e., windchill.py). You should start with file windchill.py, which can be downloaded from Brightspace. You should make a folder called "exercise_windchill" on your own computer and put file windchill.py into the folder. Then, you can open the folder in VS Code. Now you should be able to see file windchill.py in VS Code, and you are ready to add your source code into the file.

Write a function that prints the wind chill temperature for a combination of a temperature in $^\circ C$ and a wind speed in kilometer per hour. The formula to calculate the wind chill temperature (T_{wc}) is

$$T_{wc} = 13.12 + 0.6215 \times T_a - 11.37 \times V^{0.16} + 0.3965 \times T_a \times V^{0.16},$$

where parameter T_a is the temperature (measured in $^\circ C$ at 1.5 meters above the ground) and parameter V is the windspeed (in km/h on a height of 10 meters above the ground).

Use the following skeleton for `windchill.py`, in which you replace the *pass* statement with your implementation. Python keyword *pass* is a placeholder for future code; see https://docs.python.org/3/tutorial/controlflow.html#pass-statements or https://www.w3schools.com/python/ref_keyword_pass.asp

```
# GEO1000 - Assignment 0
# Authors:
# Studentnumbers:

def temp_windchill(temp_in_c, windspeed_in_kmh):
    pass

print(temp_windchill(5.0, 10.0))
```

Also, calculate the wind chill temperature for a temperature of $-1^\circ C$ and 35 km/h.

4 Note for future reference

If you want to extend Python with additional packages, you can use pip: https://pip.pypa.io/en/stable/. If you are running Windows, you can find quite some binary packages here: http://www.lfd.uci.edu/~gohlke/pythonlibs/.