Assignment 2 – Intro to Python

ECSE211 – Design Principles and Methods

(Due January 23rd, 2023, 11:59PM)

Please read this entire document before attempting the assignment.

In this course, you will use the Python programming language to write code on your Brick. The purpose of this course is not to make you a Python expert; but you will learn the basics so that you are able to implement your software design.

To make the transition from Java to Python easier, an "Intro to Python" presentation will be posted on myCourses. Additionally, a Python tutorial will be delivered by a TA to teach you some basics.

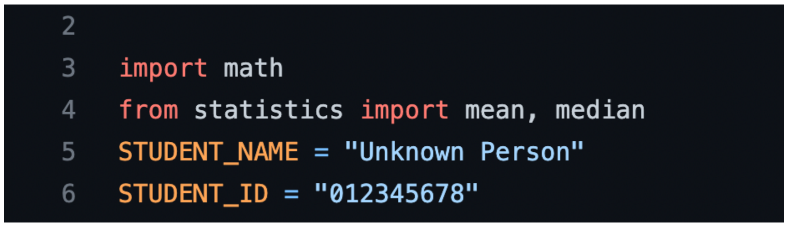
To complete this assignment, you will need to complete the following steps:

1. Download and Install Python 3.9 *(see Appendix II for instructions on installing Python)*.

2. Download and Install an IDE or code editor (i.e., VS Code, Thonny, PyCharm...). We recommend using Thonny, as this is the IDE available on the BrickPi *(see Appendix III for instructions on installing Thonny)*.

3. Download **assignment.py** and **tester.py** (available under the Assignments tab on myCourses).

4. In **assignment.py,** change **STUDENT_NAME** and **STUDENT_ID** in to match your full name and ID:

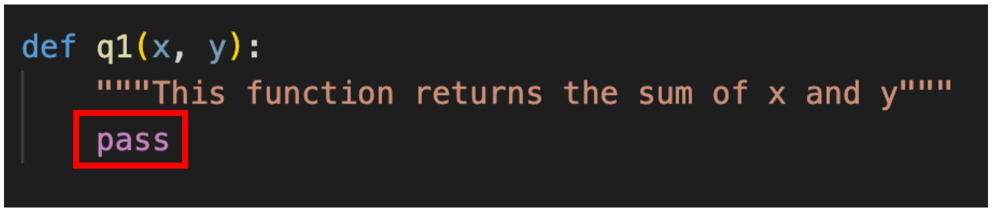


```

import math
from statistics import mean, median
STUDENT_NAME = "Unknown Person"
STUDENT_ID = "012345678"
```

5. Edit, complete, and submit **assignment.py** to MyCourses to finish the assignment.

    - To answer each question, replace the keyword **pass** in the corresponding function with your own code:



```
def q1(x, y):
    """This function returns the sum of x and y"""
    pass
```

6. **tester.py** is given to you to test your code. Put this file into the same folder as **assignment.py,** then:
    - Use the command “python tester.py” in the command terminal
    - OR execute/run the **tester.py** file in an IDE
    - OR double-click the **tester.py** file from your computer’s default file explorer

    to test your **assignment.py** file.

7. **WARNING:** ***Your code may not be fully correct even if it passes the tester file*.** You may add your own tests to the **tester.py** file to check for edge cases or validate functionality.

The assignment must be submitted through myCourses, in the Assignment 2 submission box, by **January 23rd, 2023, 11:59PM**.

Appendix I: Assignment Questions (Total : 100 points)

For each question, *r* indicates the value that the function must return.

Question 1 (3 points)

Write a function that returns the sum of x and y.

Function parameters:

Input(s):	x, y
Output(s):	r, where $r = (x + y)$

Question 2 (3 points)

Write a function that returns the product of x and y.

Function parameters:

Input(s):	x, y
Output(s):	r, where $r = (x \times y)$

Question 3 (3 points)

Write a function the returns x to the power y.

Function parameters:

Input(s):	x, y
Output(s):	r, where $r = x^y$

Question 4 (5 points)

Write a function that returns the 3 computed values in a tuple or list form:

- $2x$
- $x - 3$
- $\frac{x}{10}$ (But use integer-division, not floating-point)

Function parameters:

Input(s):	x
Output(s):	A tuple or list containing the values $2x$, $x - 3$ and $\frac{x}{10}$ in this order.

Question 5 (5 points)

Write a function to represent the following summation:

$$s = \sum_{i=x}^{y} i$$

Example: if x=1 and y=4, then the sum would be 1 + 2 + 3 + 4 = 10. Return the number 10.

Function parameters:

Input(s):	x, y
Output(s):	s

Question 6 (7 points)

Write a function to represent the following summation, while being given x as input:

$$s = \sum_{i=0}^{x} \sum_{j=0}^{5} 2 * i + j$$

<u>Hint #1</u>: You do not need to use the answer in Question 5 for this question

Example: if x=1, then the sum becomes:

(2*0 + 0) + (2*0 + 1) + (2*0 + 2) + (2*0 + 3) + (2*0 + 4) + (2*0 + 5) +

(2*1 + 0) + (2*1 + 1) + (2*1 + 2) + (2*1 + 3) + (2*1 + 4) + (2*1 + 5).

Function parameters:

Input(s):	x
Output(s):	s

Question 7 (7 points)

Create a function that returns a list of integers (of length x) where:

- The absolute value of each element is three times the index.
  For example, the absolute value of the element at index 5 should be 15.
- Values at even indices must be positive, and values at odd indices must be negative.

The output should resemble this format:

If x=5, then output = [0, -3, 6, -9, 12]

*Hint: Nothing should be printed in this question, as we ask you to **return** the list of values.*

Function parameters:

Input(s):	x
Output(s):	A list of x elements as described above.

Question 8 (7 points)

Create a function to find the index of every number 10 within the given input list.

Return these indices as a new list.

Examples:

input: [10, 9, 8, 7, 6, 10, 4, 3, 10, 1]
return value: [0, 5, 8]

input: [10]
return value: [0]

input: [3, 2, 1]
return value: []

input: []
return value: []

Function parameters:

Input(s):	lst, a list of elements that may or may not contain the integer 10.
Output(s):	A list of all indices where 10 was found in the input list

Question 9 (9 points)

Determine the GPA of the given student, using the input dict '**dct**'.
The keys of the dict are course names, while the values are the GPA obtained in each course.

Return two things as a tuple: a list of the courses taken, and the semester GPA. You can assume that all courses pull the same weight in the GPA calculation, i.e., it is sufficient to compute the average of the GPAs to determine the semester GPA.

Hint #1: 2.6 is very similar to 2.59999996 so either answer should be fine. We will use the ***math.isclose()*** function to grade this question.

Hint #2: ***dict.items()*** will be useful here. See https://docs.python.org/3.9/library/stdtypes.html#dict.items

Examples:

input: {'Calc 2' : 2.3, 'Physics' : 3.4, 'Algorithms' : 2.1}
return value: (['Calc 2', 'Physics', 'Algorithms'], 2.6)

input: {'English' : 1.5, 'History' : 4.0, 'French' : 3.8}
return value: (['English', 'History', 'French'], 3.1)

Function parameters:

Input(s):	dct, a dictionary containing the courses taken and GPAs obtained
Output(s):	A tuple containing: 1. A list of the courses taken 2. The semester GPA

Question 10 (9 points)

Create a function that performs the following operations on the given input x:

- if x less than 5 or greater than 255, return 0
- if x is between 5 and 255, return x divided by 1.25 (floating-point division)

Function parameters:

Input(s):	x
Output(s):	A single number determined by the condition specified above.

Question 11 (9 points)

Create a function that performs *vector normalization* on the 3D vector <x, y, z>

For the Normalized 3D Vector *u*, given the input vector *v*, the formula is:

$$u = \frac{v}{|v|},\ |v| = \sqrt{x^2 + y^2 + z^2}$$

Return a list or tuple of the 3 normalized x, y, z values

Hint #1: Use the **math** python module

Hint #2: Vector normalization is the process of transforming a vector into a unit vector while preserving its original direction.

Function parameters:

Input(s):	x, y, z
Output(s):	A tuple containing the normalized x, y, z values.

Question 12 (9 points)

Write a function to convert Polar Coordinates into Cartesian Coordinates.

<u>Hint:</u> Both functions **math.cos()** and **math.sin()** USE RADIANS. See https://docs.python.org/3.9/library/math.html for available angular conversion functions.

Examples:

input: [10, 60]
answer (not exact): (5.000000000000001, 8.660254037844386)

input: [10, 30]
answer: (8.660254037844387, 4.999999999999999)

input: [1, 180]
answer: (-1.0, 1.2246467991473532e-16)

input: [13, 22.6]
answer: (12.001732822468751, 4.9958391945774485)

Function parameters:

Input(s):	$radius, degrees$, corresponding to the radius and angle (in degrees) of the polar coordinates.
Output(s):	A tuple containing the cartesian coordinates x, y of the input.

Question 13 (12 points)

From a 2D list of numbers, give the Minimum, Maximum, Mean, and Median values of each row.

The format of these lists is as follows:

```
[
        [1,2,3,4,5],
        [3,4,5],
        [45,83,65,52,47]
]
```

The number of give lists is variable, and the length of each list is variable. Your code will not be tested for speed performance.

Output your data as a string. Each list's stats are in one row, which is separated by a newline character.

Each number is separated by a comma, such as this:

```
1,2,3,4
6,7,8,9
1,2,3,4
3,6,8,9
```

Example:

```
input: [[1, 3, 5, 7], [2, 4, 6, 8], [3, 5, 7, 9], [4, 6, 8, 10]]
return value:
        "1,7,4,4.0
         2,8,5,5.0
         3,9,6,6.0
         4,10,7,7.0"
```

*Note: Do not use **print().** Create and build the string in a variable, then return the string.*

Do not add extra spaces. You may end each row with a newline. (Your testing program will expect this).

Function parameters:

Input(s):	A 2D list of numbers
Output(s):	A string of comma-separated values as described above.

Question 14 (12 points)

Take the following string input and convert it into a 2-dimensional list.

Example:

```
Input:
1,2,3
4,5,6,7,8
9,102,5,25,3

Output:
[
  [1,2,3],
  [4,5,6,7,8],
  [9,102,5,25,3]
]
```

Requirement:

- The numbers within the 2D list output must be of the int type, not the str type.

Rules for Input Data:

- The input string will not have any spaces in it.

- Each row is separated by a newline character.
- There is no newline character at the end of the Input String (i.e., at the end of the last row).
- Each number is separated by a comma.
- The numbers will all be integers within -2147483648 to 2147483647.

Function parameters:

Input(s):	A string of comma-separated values as described above.
Output(s):	A 2D list of numbers extracted from the string.

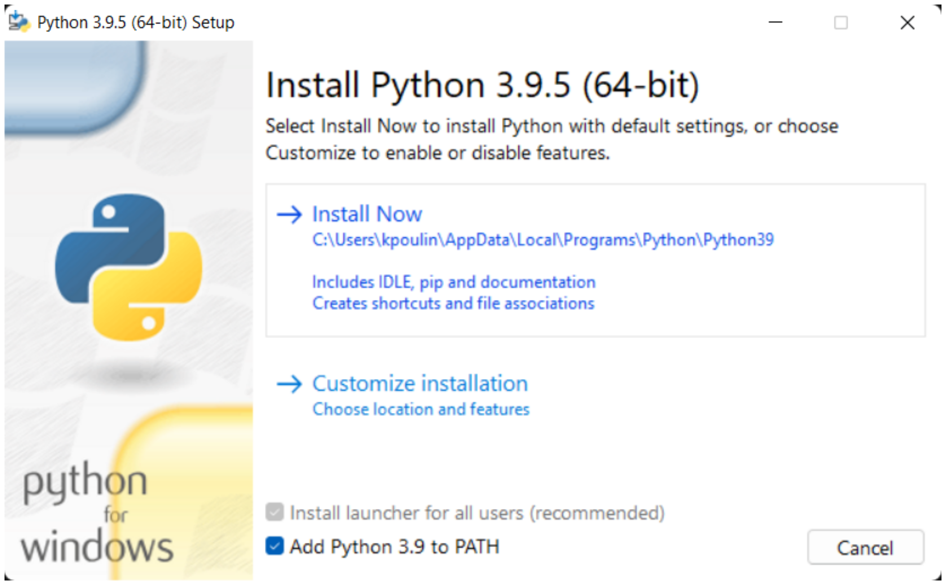
Appendix II: Python Environment Installation Instructions

For this course you will need Python 3.9 or higher. Download the installer here: Python 3.

Select the appropriate installer for your operating system.

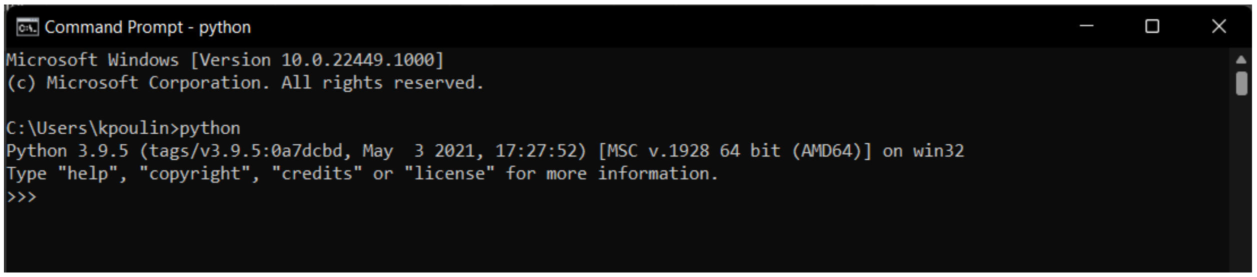
Windows instructions

1) Double-click on the installer to run it. You should get the following window:



Python 3.9.5 (64-bit) Setup
Install Python 3.9.5 (64-bit)
Select Install Now to install Python with default settings, or choose Customize to enable or disable features.
Install Now
C:\Users\kpoulin\AppData\Local\Programs\Python\Python39
Includes IDLE, pip and documentation
Creates shortcuts and file associations
Customize installation
Choose location and features
python for windows
Install launcher for all users (recommended)
Add Python 3.9 to PATH
Cancel

2) Make sure to check the “Add Python 3.9 to PATH” box and click *Install Now*.
3) After getting the “Setup was Successful” message, you can close the window.
4) Test your installation of Python 3.9 by opening a command line terminal and typing python. You should get the following response:



```
Microsoft Windows [Version 10.0.22449.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kpoulin>python
Python 3.9.5 (tags/v3.9.5:0a7dcbd, May  3 2021, 17:27:52) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

macOS Instructions

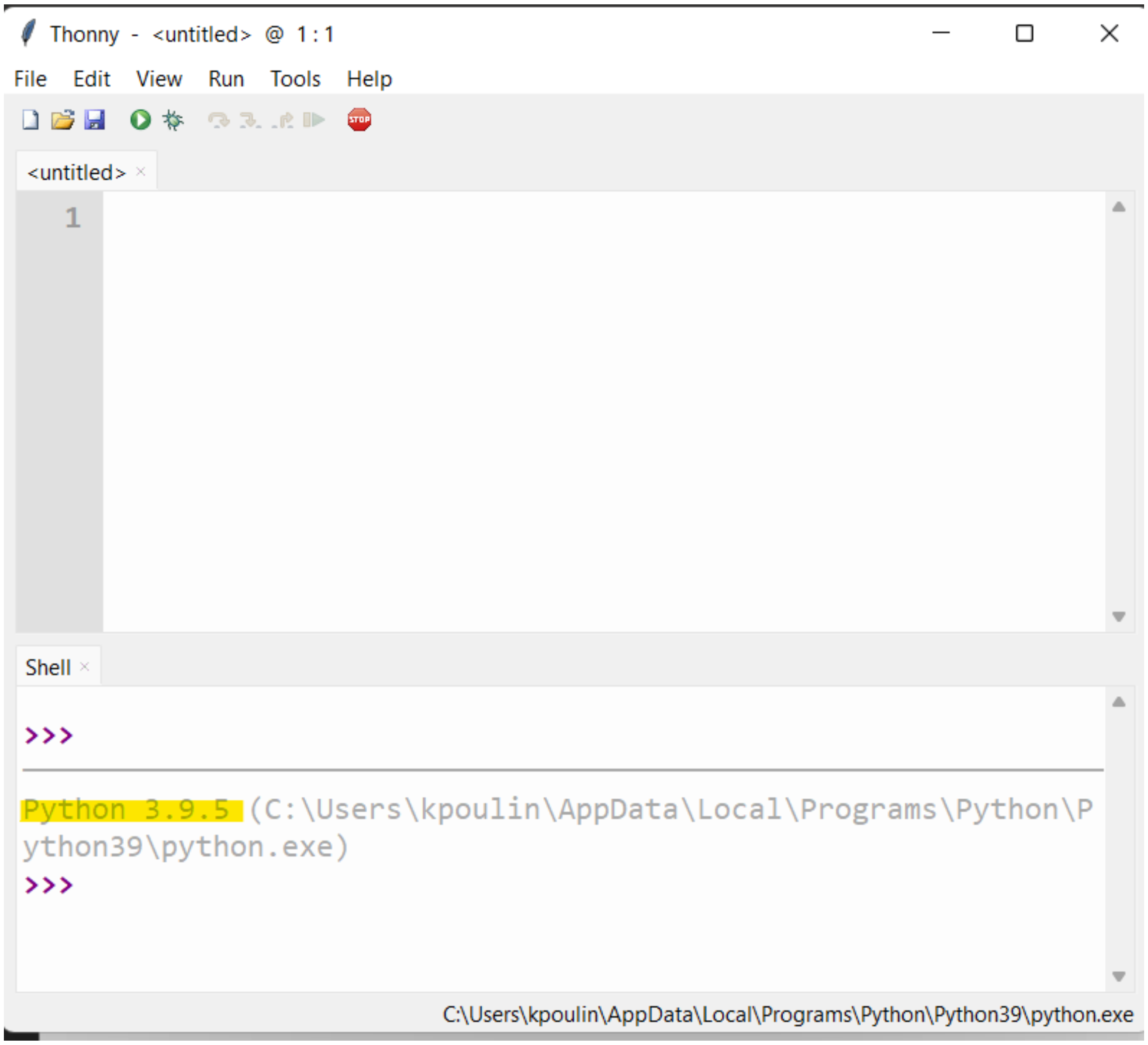
1) Run the installer. You do not need to change any default installation settings.
2) Similarly to Windows, writing python in a terminal should lead you to the command-line interpreter.

Appendix III: Thonny Installation Instructions

Thonny is a Python IDE available on Windows, MacOS and Linux. If you have a preferred IDE that you know how to use, eg, Visual Studio Code, you can use it for the assignment. However, if you do not have a Python IDE on your computer, Thonny is the way to go since it is also used on the BrickPi.

You can download Thonny using this link: Thonny, Python IDE for beginners.

Once it is downloaded, open Thonny. Make sure you are running Python 3.9 or higher here (see highlighted part of screenshot):

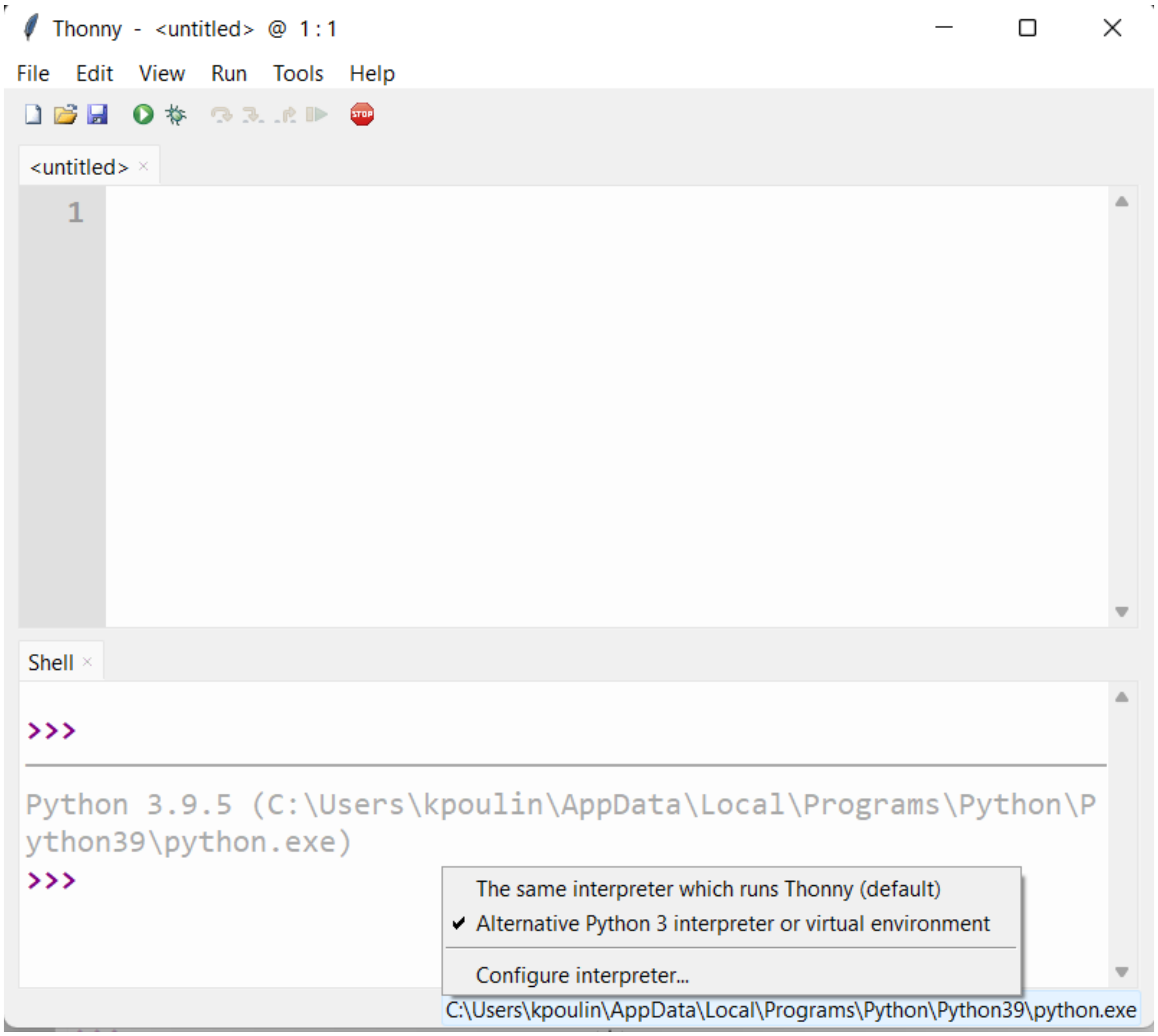


```
>>>

Python 3.9.5 (C:\Users\kpoulin\AppData\Local\Programs\Python\P
ython39\python.exe)
>>>
```

C:\Users\kpoulin\AppData\Local\Programs\Python\Python39\python.exe

If this is not the case, click on the Python version on the bottom-right corner of the window (picture above) and select "Alternative Python 3 interpreter or virtual environment".



```
>>>
Python 3.9.5 (C:\Users\kpoulin\AppData\Local\Programs\Python\Python39\python.exe)
>>>
```

The same interpreter which runs Thonny (default)
✓ Alternative Python 3 interpreter or virtual environment
Configure interpreter...

C:\Users\kpoulin\AppData\Local\Programs\Python\Python39\python.exe

The Python interpreter at the bottom has the same functionalities as the command prompt window shown earlier (in the Python installation instructions).

You can open the assignment template file by clicking *File > Open* and selecting the template file. You can then edit the assignment template file directly from the editor.