Assignment 2 – Intro. to Python

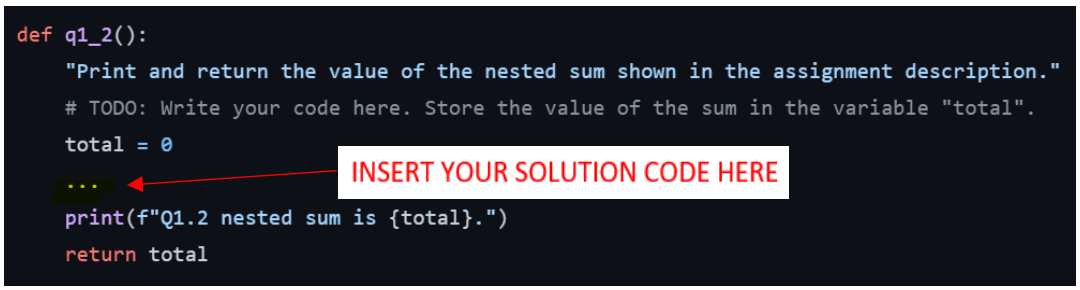
ECSE211 – Design Principles and Methods

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 design.

To make the transition from Java to Python easier, a Python for Java Programmers Guide is available here. Make sure to review this guide before attempting this assignment.

To complete this assignment, you should download the *ECSE211_A2_StudentID.py* file in the Assignment-2 folder on myCourses. Rename this file with your own 9-digit student ID, eg, *ECSE211_A2_123456789.py*. This file contains a template for submitting the assignment, question by question. This file will be your only submission for this assignment.

For each question, you are expected to replace the "..." characters with your solution code, like this:



You will be penalized if you do not follow the template provided in the .py file.

You will also need to install Python (and Thonny) on your local machine. Instructions for Windows, MacOS and Linux are provided below.

Python Environment Installation Instructions

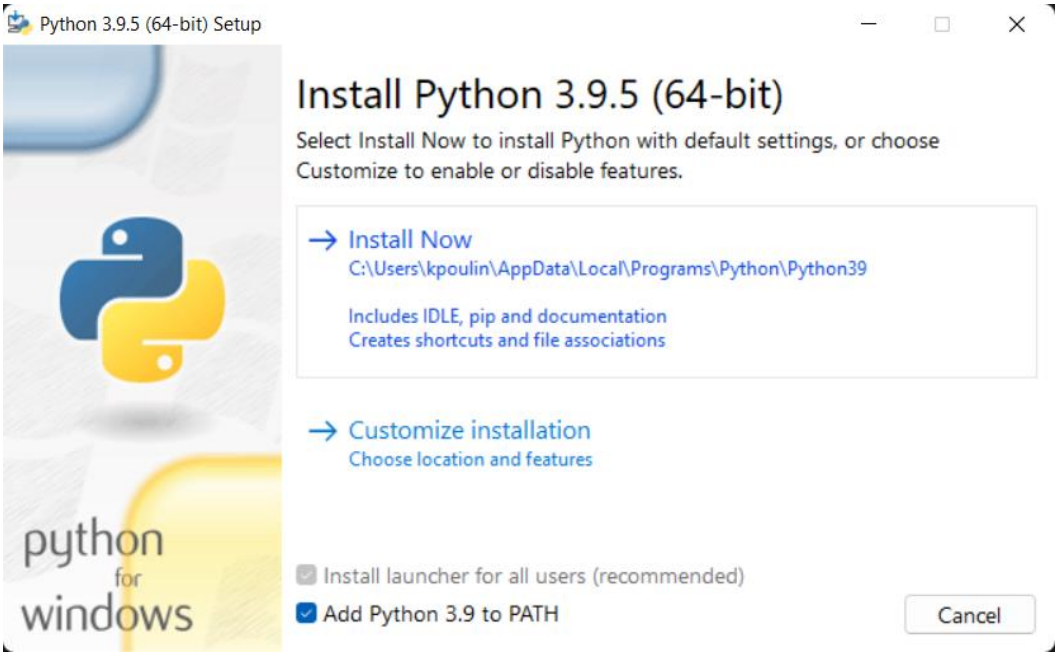
For this course you will need Python 3.9. Download the installer using the following link: Python 3.9.5.

Scroll down to Files, then select the appropriate installer for your operating system:

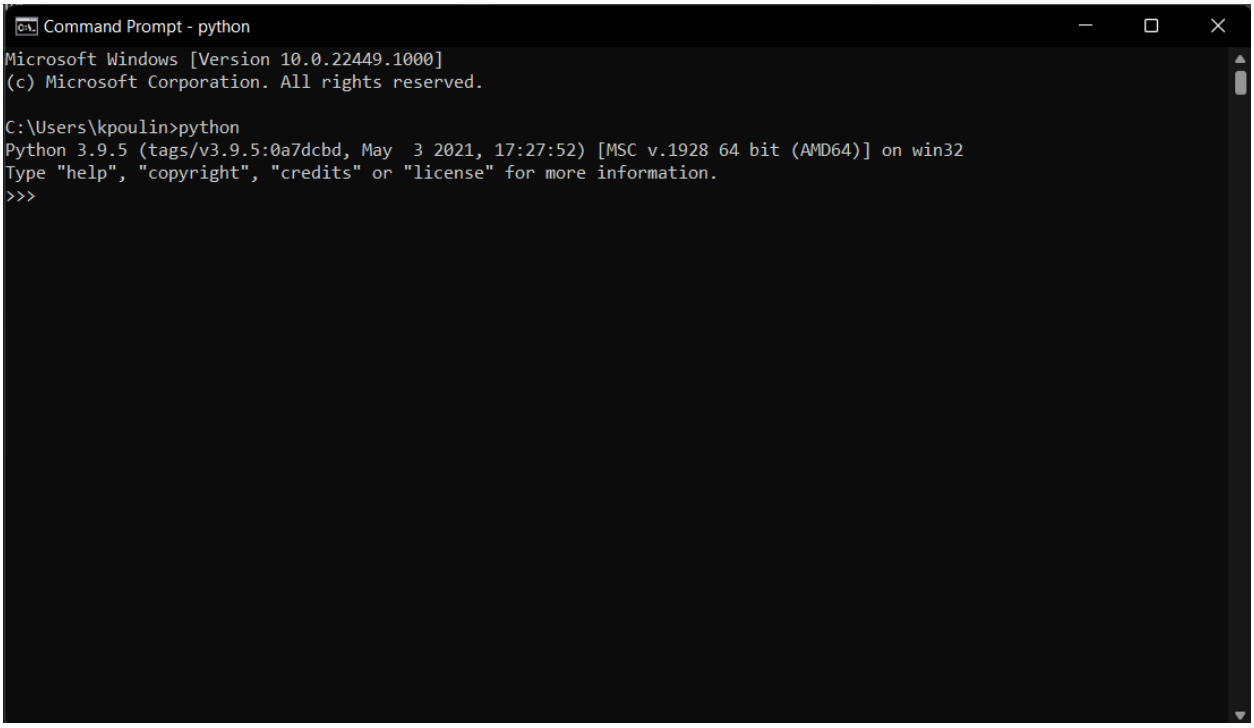
a. Windows: *Windows installer (32-bit)*
b. MacOS: *macOS 64-bit universal2 installer*

Windows instructions

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



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:



MacOS Instructions

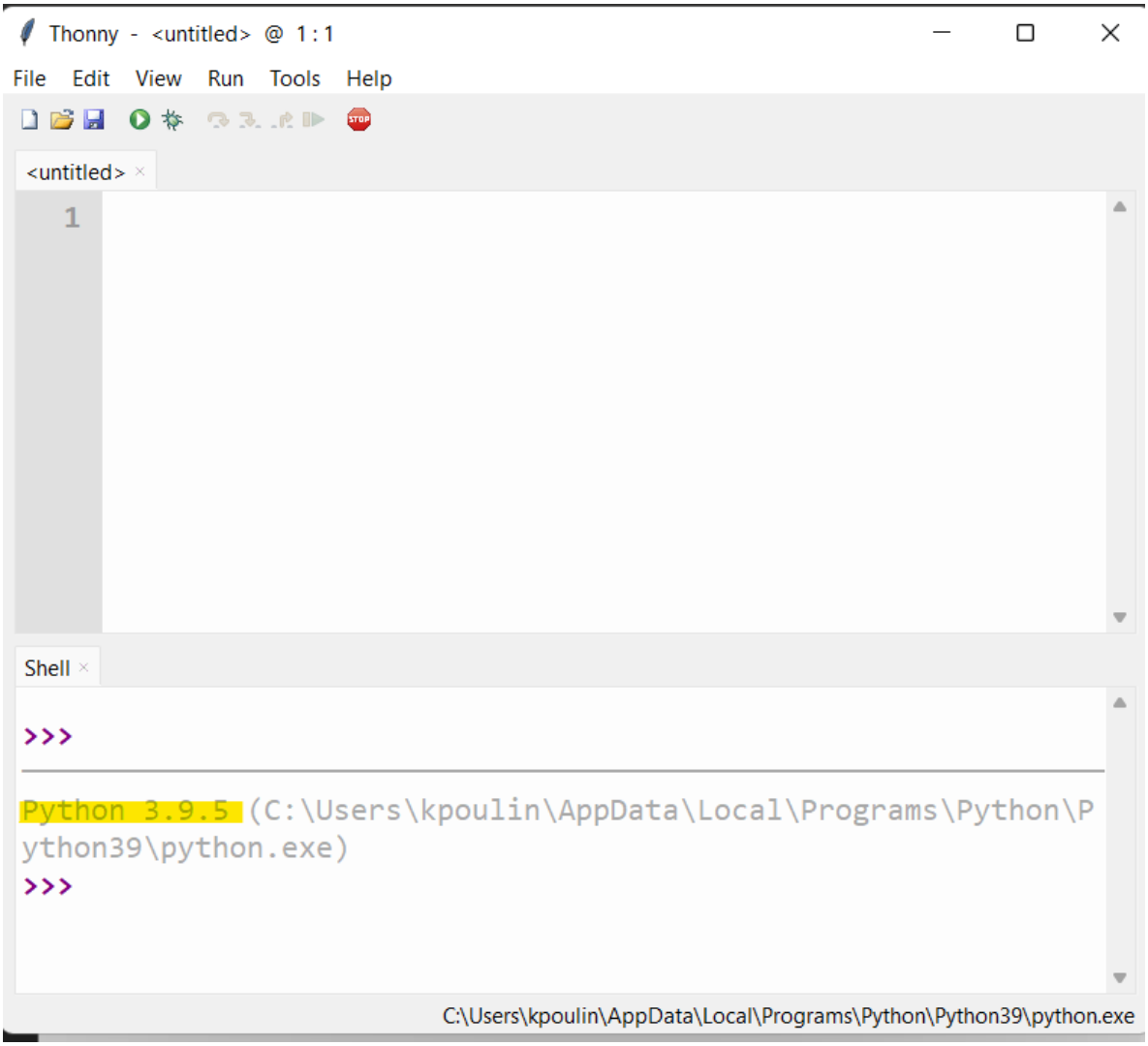
1) Run the installer. You do not need to change any default installation settings.
2) Similarly to Windows, writing python in a terminal should lead you to the command-line interpreter.

Thonny Installation

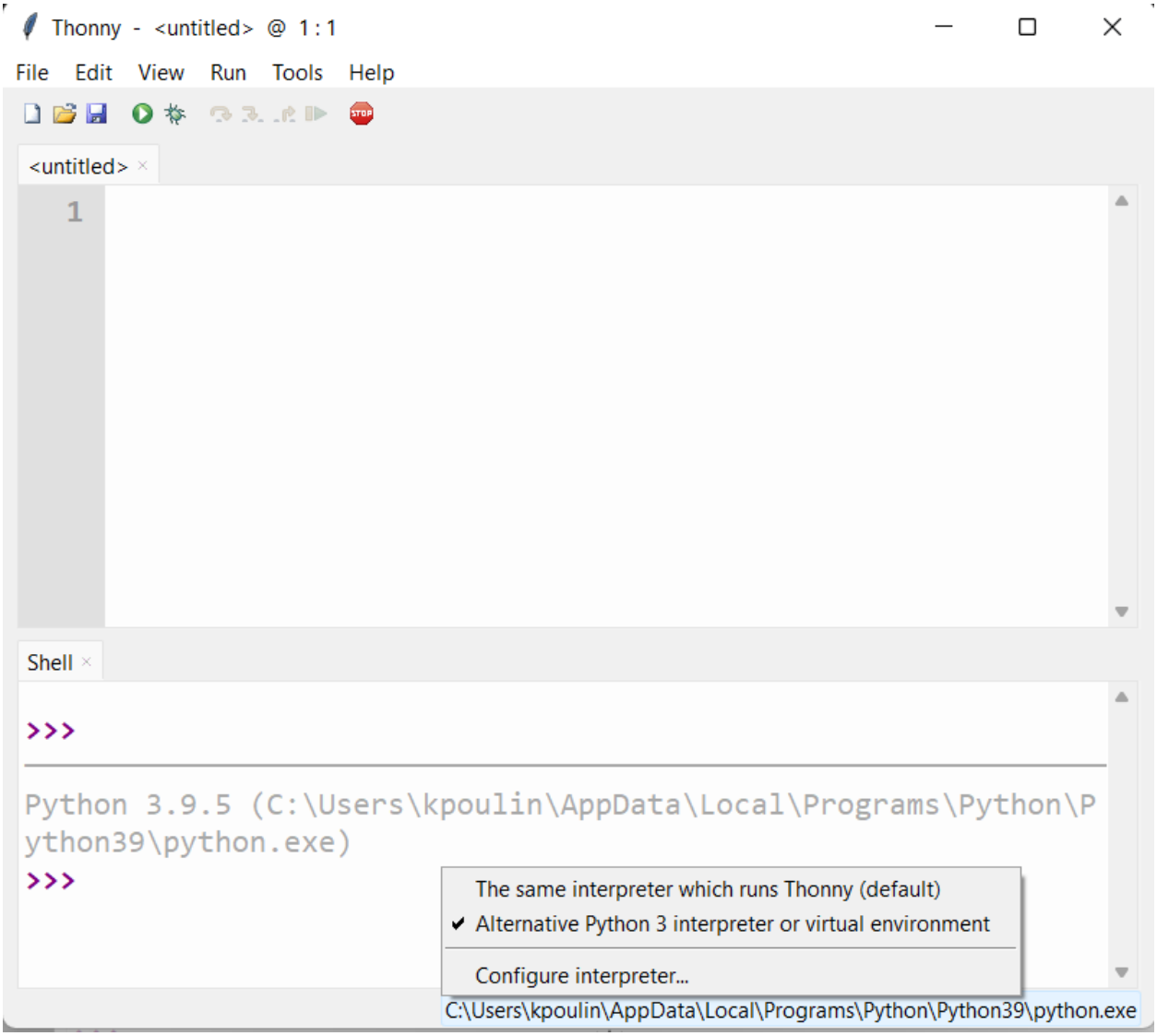
Thonny is a Python IDE available on Windows, MacOS and Linux. If you have a preferred IDE that you know how to use, you can use it for the assignment. However, if you do not have a Python IDE on your computer, Thonny is the way to go since it is also the IDE we will be using on the BrickPi.

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 here:



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”



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.

Test your answers by running the given file (use the play button). You are free to modify the testing code at the bottom of the file since it will not be graded.

Assignment Questions

Part 1 - Review of basic tutorial concepts

In this part, you will answer several short questions related to the Python for Java Programmers document. Please go over it if you have not already done so.

Question 1.1 (1 point)

Enter a Python expression that represents the sum of 2 and 2.

Question 1.2 (8 points)

Write Python code that prints the value of this nested sum:

$$\sum_{i=0}^{20}\sum_{j=0}^{21}(2i+j)$$

Question 1.3 (8 points)

Create and print a list of integers of length 21, where:

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

Hint: The first few values will look like this: [0, -3, 6, -9, ...]

Question 1.4 (8 points)

Find the average (mean) hours that each student spends on one of their courses (every week) and print it in a `dict` format similar to the inputs (copy/paste this code below):

```
ecse200_hours: dict[str, float] = {"Alpha": 3.5, "Bravo": 6.0, "Charlie": 5.5}
ecse202_hours: dict[str, float] = {"Alpha": 4.0, "Bravo": 3.5, "Charlie": 2.0}
math141_hours: dict[str, float] = {"Alpha": 5.0, "Bravo": 3.0}  # Charlie does not
take this course
```

Question 1.5 (15 points)

Implement a general version of the algorithm from Q1.4 that takes in the following input:

- A list of `dicts`, each one representing weekly hours spent on this course by each student in the exact same format as Q1.4. Note that the class lists for different courses can differ.

The output must be a `dict` with all students and the average time they spend on one of their courses. If a student did not take a course, do not assume they spend 0 hours on the course for the course (e.g., Charlie's average (mean) from the example above is

$$avg_{Charlie} = \frac{\sum Charlie's\ hours}{Num\ courses\ he\ took} = \frac{2.0 + 5.5}{2} = 3.75$$

Hints: In general, your algorithm should first find all the student names. Then, like before, use a nested for loop where you iterate over the students and then over each course they took.

Part 2 – Find the bugs

Each code snippet in the template contains one or more bugs (mistakes). Identify the bugs and fix them to make the code run as intended.

Question 2.1 (2 points)

The code provided in the template should print the sum of numbers from 0 to 100, inclusive.

Question 2.2 (3 points)

The code provided in the template should modify a string by replacing all references to "2020" with "2021".

Question 2.3 (5 points)

The code provided in the template should find the three smallest even numbers in a list.

Question 2.4 (5 points)

The code provided in the template adds the colors "orange" and "red" to a rainbow colors dictionary, then prints the values in the dictionary.

Part 3 - Code documentation (10 points)

In this part, enter an explanation wherever there is a ***"..."****. The code you need to document is already in your submission file.*

Note: To make the above code easier to understand, meaningful variable/function names could have been used in addition to code comments. That is what you are required to do in all the code you write, except for obfuscation contests and the like.

Part 4 - Translation from Java (35 points)

In this part, translate the given Java code into code that does the same thing in Python.

```java
/** Represents a coordinate point on the playing field grid. */
      public class Point {
        public double x; // The x coordinate in tile lengths.
        public double y; // The y coordinate in tile lengths.
        private static final double EPSILON = 0.003; // In ft. The threshold for coordinates to be
considered equal.

        /** Constructs a Point. The arguments are in tile lengths. */
        public Point(double x, double y) {
          this.x = x;
          this.y = y;
        }

        /** Returns the distance between the two points in tile lengths (feet). */
        public double distanceTo(Point other) {
          double dx = other.x - x;
          double dy = other.y - y;
          return Math.sqrt(dx * dx + dy * dy);
        }

        /** Makes points from a string in a comma-separated format, eg "(1,1),(2.5,3)". */
        public List<Point> makePointsFromString(String s) {
          List<Point> result = new ArrayList<Point>();
          if (s == null || !s.contains(")")) {
            return result;
          }
          s = s.replaceAll("\\s+", "").replaceAll("\\(", "").replaceAll("\\),", "\\)");
          for (String fragment: s.split("\\)")) {
            String[] xy = fragment.split(",");
            result.add(new Point(Double.parseDouble(xy[0]), Double.parseDouble(xy[1])));
          }
          return result;
        }

        @Override public boolean equals(Object o) {
          if (!(o instanceof Point)) {
            return false;
          }
          Point other = (Point) o;
          return Math.abs(x - other.x) < EPSILON && Math.abs(y - other.y) < EPSILON;
        }

        @Override public String toString() {
          DecimalFormat fmt = new DecimalFormat("#.##");
          return "(" + fmt.format(x) + ", " + fmt.format(y) + ")";
        }
      }
```