CS 3035, Fall 2022
In-Lab Exercise 12

UPDATED — Due: ~~September 30,~~ October 2, 2022 (11:59 PM midnight)

Testing Python Typing rules and capabilities

Today's lab is divided into two parts. In Part 1, you will do the lab exercises stated here individually. This part will last from 2:50 - 3:30 PM. In Part 2, you will be assigned to breakout rooms where you will discuss your observations with your assigned group. This will last from 3:30 PM - 3:45 PM. In Part 3, we will have a short class-wide discussion on interesting observations you made during your individual and discussion portions.

1. Experiment with different data types using the Python interpreter shell. Create a data object of each type: number, string, list, dictionary. Check the data type shown for each type. Take a screenshot of your answer.

2. Python uses expression operators +, -, *, /, **. These mean different things depending on data types.
   A. Check which of these expression operators apply to numbers vs. strings. What is the difference in what they do for each data type? Write your observations.
   B. Check the typing rules that apply to these expressions when you apply them to expressions containing numbers and strings. Write your observations.

3. Python includes methods that are applicable to objects of different data types. For a list L or a string S in Python, you can list the object methods available using `dir(L)` or `dir(S)`. More details of each method can be found using the `help()` function. The help function accepts a data type as an argument and returns all associated in-built methods. To get out of the help() function, press 'q'. Make sure you use the data type name as listed in the output of type() as an argument to the help function

   A. Note down at least 3 methods of your choice for strings, dictionaries, and lists. You can ignore methods with leading and trailing double underscores. These are used to represent implementations of objects and support customization. The names without underscores are callable methods.
   B. What do you think these methods should do? Write in 1-2 lines for each method.
   C. Do some quick testing in the Python interpreter shell to see if your methods of choice did what you expected them to do. Write your observations and show a screenshot of applied methods.

4. Python also includes modules like the Math module and the Random module. To load the math module, do `import(math)`. Use `dir(math)` to check the methods within the Math module. Math capabilities can be run as math.<method_name>. For example, the Math module includes a 'pow' method. To use this method, you can run `math.pow(x,y)`where x and y are the numbers of your choice. To find out more about a method, use the help function. For example, `help(math.pow)` tells you what the pow function in the math module does.

   A. Note down 5 methods within the math module and write down what you think these methods should do.

B. Do some quick testing in the Python interpreter shell to see if your methods of choice did what you expected them to do. Write your observations and show a screenshot of applied methods.
C. Observe what dynamic typing / strongly typing rules apply to different types. For example, try using the math module methods on different data types and check the type of the result. Write down a short summary of your observations. We will discuss these observations in class.

5. Strings are immutable while lists and dictionaries are mutable. Do some quick testing in the Python interpreter shell to test this claim. Does this hold true for the cases you tried? Show a screenshot of your results.