CS 3035, Fall 2022
In-Lab Exercise 11

Due: September 28, 2022 (11:59 PM midnight)

Examining Python's dynamic typing

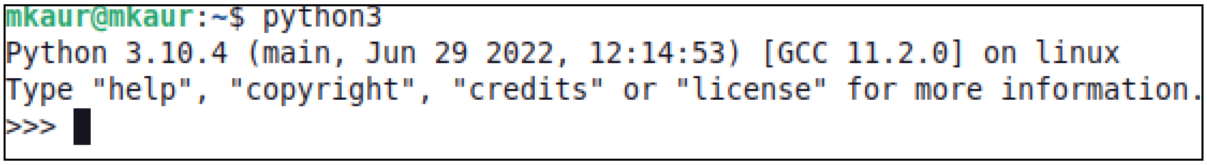
*In today's lab exercise, we will examine the characteristics of Python's dynamic typing using the Python Interactive Shell . Please follow the following lab instructions and answer the questions in red. Your answers should be in a single PDF file. Each response should be clearly numbered with the corresponding question number. Please make sure that your responses are in the format required for the questions. Each question states the type of response expecte*d.

Python has two basic modes: script and interactive. The normal mode is the mode where the scripted and finished .py files are run in the Python interpreter. Interactive mode is a command line shell which gives immediate feedback for each statement, while running previously fed statements in active memory. As new lines are fed into the interpreter, the fed program is evaluated both in part and in whole. Interactive mode is a good way to play around and try variations on syntax.[1] **To exit out of the interactive shell at any point, either use CTRL-Z or type quit() and enter.**

Lab Instructions:

1. Start an interactive Python session by typing the following in your Virtual Machine's terminal. `python3` or `python3.10`

   This should result in an output similar to the following screenshot:



```
mkaur@mkaur:~$ python3
Python 3.10.4 (main, Jun 29 2022, 12:14:53) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 
```

   The >>> is Python's way of telling you that you are in interactive mode. In interactive mode, what you type is immediately run. Try typing 1+1 in. Python will respond with 2. Interactive mode allows you to test out and see what Python will do. If you ever feel the need to play with new Python statements, go into interactive mode and try them out.[2]

2. The following program lines, written and executed in the Python interactive shell, accepts variables of different data types and performs operations on them. Please observe the behavior of Python after each line is input.

   2.1. Type in `var1 = 5` and press ENTER.

[1] https://en.wikibooks.org/wiki/Python_Programming/Interactive_mode

[2] https://en.wikibooks.org/wiki/Python_Programming/Interactive_mode

2.2.To check the data type of a, type in `type(a)` and press ENTER.

2.2.1. What is the data type of the variable ‘a’ ? State your answer in your own words and paste a screenshot of your output.

2.2.2.Notice that the interpreter knows the data type without us specifying it. What does this indicate about the Type Binding that Python uses? State your answer in <1 line.

2.3. Let us introduce a second variable ‘var2’ now. Assign value to var2 by typing `var2 = var1*5.65` and press ENTER. We can check the value assigned to var2 by typing `print(var2)`

2.3.1.What is the value assigned to variable ‘var2’ ? State your answer in your own words and paste a screenshot of your output.

2.3.2.What is the data type of variable ‘var2’ ? Use the command from 2.2 to check the data type. State your answer in your own words and paste a screenshot of your output.

2.3.3.Notice that ‘var1’ and ‘var2’ have different data types. What characteristic of data type conversions in Python does this behavior indicate? State your answer in <1 line.

2.4.Let us introduce a third variable ‘str1’ now. This variable will hold a string value. Assign value to str1 by typing `str1 = 10` and press ENTER.

2.4.1.What is the data type of variable ‘str1’ ? Use the command from 2.2 to check the data type. State your answer in your own words and paste a screenshot of your output.

2.5.Let us try to perform an arithmetic addition operation adding str1, var1, and var2. Type in `str1 + var1 + var2` and press ENTER.

2.5.1.What is the output that you observe? State your answer in your own words and paste a screenshot of your output.

2.5.2.Which activity performed by the interpreter results in this output?

2.6.Python also supports explicit typecasting where we can type cast one data type to another. To typecast an int ‘var1’ to a string, we can use (str)int1 or “int1”. Let us try to make the following changes to the arithmetic edition expression we were using in 2.5. Type in `(int)str1 + var1 + var2` and press ENTER.

2.6.1. What is the output that you observe? State your answer in your own words and paste a screenshot of your output.

2.6.2.Why did you not see an error this time?

2.7. Now type in `str1 + (str)var1 + (str)var2` and press ENTER.

2.7.1. What is the output that you observe? State your answer in your own words and paste a screenshot of your output.

2.7.2.What operation does this expression seem to be doing? State your answer in <1 line.