



Linear Algebra

Laboratory Activity No. 1

Python Programming

Submitted by:

Marquita, Bhea Marrienne J.

Instructor:

Engr. Dylan Josh D. Lopez

February 21, 2021

I. Objectives

This laboratory activity aims to implement the principles and techniques of the basic Python programming language. This laboratory activity aims to establish or review the basic skills in Python programming. This laboratory activity also seeks to create a program that computes the semestral grade in two decimal places, input and display their name and course, and displays an emoji based on how high the student's semestral grade is.

II. Methods

The researcher reviewed Python programming fundamentals such as Variables and Data Types, Operations, Input and Output Operations, Logic Control, Iterables, and Functions.

- **Variables and Data Types**

In Python, some names/codes can't be used as variables, such as and, is, or, etc. These codes shouldn't be used because it will cause a complication if you run the program.

In Python, you can declare multiple variables in one line, and it is by using comma [,], for example: b, h, e, a = 1, 2, 3, 4. Moreover, there's a built-in function in Python wherein you can check the variable's data type. On the example given, you can check the data types of each variable using the function type(b), type(h), etc., and it will display the data type, and based on the examples, they're all integers [int]. Additional information is that when you declare a value, it will only display the last line of code in a cell.

Python is like any programming language that has various data types, and they are the following:

Test Type	str
Numeric Types	int, float, complex
Sequence Types	list, tuple, range
Mapping Type	dict
Set Types	set, frozenset
Boolean Type	bool
Binary Types	bytes, bytearray, memoryview

○ Operations

In Python, there are seven arithmetic operators that are used to perform mathematical operations namely Addition [+], Subtraction [-], Multiplication [*], Division [/], Modulus [%], Exponentiation [**], and Floor Division [//].

Assignment operators are used in Python to assign values to variables. [1] Here are the assignment operators used in the Python programming language:

Operator	Example	Result
=	d = 4	d = 4
+=	d += 4	d = d+4
-=	d -= 4	d = d-4
*=	d *= 4	d = d*4
/=	d /= 4	d = d/4
%=	d %= 4	d = d%4
//=	d //= 4	d = d//4
=	d **= 4	d = d4
&=	d &= 4	d = d&4
=	d = 4	d = d 4
^=	d ^= 4	d = d^4
>>=	d >>= 4	d = d>>4
<<=	d <<=4	d = d<<4

Comparison operators are used to comparing values. It returns either True or False according to the condition. [2] Here are the comparators used in Python:

Operator	Meaning
>	Greater than
<	Less than
==	Equal to
!=	Not equal to
>=	Greater than or equal to
<=	Less than or equal to

Logical operators are simple and, or, and not. This is what the researcher is learning about under the subject course Discrete Mathematics, so they're kind of familiar with it. Logical operators also include a truth table which looks complicated at first.

- **Input and Output Operations**

Python provides numerous built-in functions that are readily available to us at the Python prompt. The first function, which is also the most common operator in Python, is `print()`, wherein the user can output the text inside the parenthesis and quotation marks. The second operator introduced is `count [cnt]`, it searches the substring in the given string and returns how many times the substring is present. It also takes optional parameters to start and end to specify the starting and ending positions in the string, respectively. [3]

Looping statements are continuous running loops that are given a certain condition and if that condition was met, the loop will end. [4] The first type of looping statement is the while loop, it repeats a statement or group of statements while a given condition is True. It tests the condition before executing the loop body. The second type of loop statement is for which executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. [5]

- **Flow Control**

Condition statements in Python are if, if-else, and elif statements. If statement is one of the most commonly used conditional statements in out of all programming languages. It decides whether certain statements need to be executed or not. If statement checks for a given condition, if the condition is true, then the set of codes present inside the if block will be executed. While if-else statement tells that if a given condition is true then execute the statements present inside if block and if the condition is false, then execute the else block. And lastly, the elif statement is used to check multiple conditions only if the given condition is false. It's similar to an if-else statement, and the

only difference is that in else, we will not check the condition, but in elif we will do check the condition. [6]

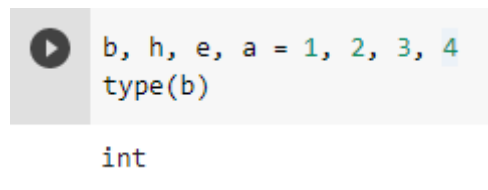
- **Functions**

Functions are declarations inside the code that run a specific routine every time it's been called. Functions could be parametric and non-parametric. Functions that are parametric require parameters, while non-parametric functions don't require any parameters, they're like void functions. [4]

The researcher also created a program wherein the objectives are explicitly met: a program that computes a student's semestral grade in two decimal places, inputs and displays their name and course, and displays an emoji based on how high the student's semestral grade is.

III. Results

A. Python Fundamentals

A screenshot of a Python code editor. On the left, there is a play button icon. The code consists of two lines: `b, h, e, a = 1, 2, 3, 4` and `type(b)`. Below the code, the output `int` is displayed. The code is highlighted in a light blue box, and the output is in a light green box.

```
b, h, e, a = 1, 2, 3, 4
type(b)

int
```

Figure 1.0 Variables and Data Types

In Figure 1.0, we can see the practice version of the researcher wherein they first declared the values of the variables b, h, e, a, then used the built-in function of Python to know its data type.

```
[4] ## Addition
    b, h, e, a = 1, 2, 3, 4
    c = b+h
    c
```

3

```
[5] ## Subtraction
    l = e-h
    l
```

1

```
[6] ## Multiplication
    u = a*b
    u
```

4

```
[7] ## Division
    m = a/h
    m
```

2.0

Figure 2.0 Arithmetic Operators

The researcher tried the arithmetic operators on Google colab using the same variables and values from Figure 1.0. They used four of the seven arithmetic operators, namely Addition [+], Subtraction [-], Multiplication [*], and Division [/].

Assignment Operators

```
[10] ## =  
a, s, h, t, o, n = 19, 94, 7, 7, 12, 3
```

```
[11] ## +=  
h = h+a  
h += a  
h
```

45

```
[12] ## -=  
t = n*s  
t -= o  
t
```

270

```
[14] ## /=  
s = h/t  
s /= t  
s
```

0.0006172839506172839

Figure 2.1 Assignment Operators

The researcher also practiced using the assignment operators, as shown in Figure 2.1. It should be note taken that a user must declare the value of a variable to avoid complication and confusion.

```
[16] l, u, k, e = 19, 96, 7, 16  
      true_val = 16
```

```
[17] ## Equality  
      e == true_val
```

True

```
[18] ## Non-equality  
      k != true_val
```

True

```
[19] ## Inequality  
      l <= true_val
```

False

Figure 2.2 Comparison Operators

Comparison Operators or Comparators are primarily used when comparing a value of a variable to another variable. As shown in Figure 2.2, the researcher used comparators to know if the values of each condition.

```
## while loop  
m, i = 10, 21  
while(m<i):  
    print(m)  
    m+=1
```

10
11
12
13
14
15
16
17
18
19
20

Figure 3.0 While Loop

The researcher tried using the while loop and (as shown in Figure 3.0) the operation was successful because the variable m started with 10 value and when it reaches the value of 20, it stopped because the condition given is less than [<] which means that it must display a value less than the value of the variable i.

B. Grade Calculator

Please refer to https://github.com/bheanne/LinearAlgebra/blob/main/LAB%20REPORT%201/Marq_Lab1.py for the codes of the program.

```
1 def grades_average(name, course, prelims, midterms, finals, border):
2
3     final_average = (float(prelims) + float(midterms) + float(finals))/3
4     final_grade = round(final_average, 2)
```

Figure 4.0 Declaration and Formula

On the program created, the researcher declared `grades_average(name, course, prelims, finals, border)` using the `def` function in Line 1. At the Line 3, the researcher inputted the formula for the average of the semestral grades using a float data type. Lastly, on Line 4, the researcher used the `round` function to round the average to the nearest two decimals to meet this laboratory activity's aim.

```
6 Happy = ("\U0001F600 :>> " + "Congrats on getting a high grade!")
7 Laughing = ("\U0001F606 :| " + "Congrats on getting a passing grade!")
8 Sad = ("\U0001F62D :<< " + "Never give up!")
9
10 if final_grade > 70.00:
11     emoji = Happy
12 elif final_grade == 70.00:
13     emoji = Laughing
14 else:
15     emoji = Sad
```

Figure 4.1 Emojis

From Line 6 until Line 15, it was all connected to one of the program's objectives: to display an emoji as feedback for how high the students' grades are. In Line 6-8, the UniCode

[illegible]

From Line 17 until Line 30, the researcher used the Input and Output operations which are necessary for the program to meet its objectives.

From 1st year 1st semester until today, the researcher had already learned three programming languages, namely, C++, C#, and Python, and only found out today that it is essential to know more than one language because it can be used after graduation, especially when a program had to be translated from Python to a more basic language like C++.

9

emojis just by inserting Unicodes (and there are two more ways based on what the researcher has read) and was also able to discover that special characters can be used as borders too. This laboratory activity made the researcher want to learn and explore more.

References

- [1] (2021) w3schools.com. [Online].
https://www.w3schools.com/python/gloss_python_arithmetic_operators.asp
- [2] (2020, September) Programiz. [Online]. <https://www.programiz.com/python-programming/operators>
- [3] (2020) Programiz. [Online]. <https://www.programiz.com/python-programming/methods/string/count>
- [4] Dylan Josh Domingo Lopez. (2021, February) Google Drive. [Online].
<https://drive.google.com/drive/u/7/folders/1i-BqIG09TqEDSUeNakWoWQ0EWqevS83E>
- [5] (2021) w3schools.com. [Online].
https://www.w3schools.com/python/python_for_loops.asp
- [6] (2021, March) Software Testing Help. [Online].
<https://www.softwaretestinghelp.com/python/python-conditional-statements/>
- [7] (2020, August) Programiz. [Online]. <https://www.programiz.com/python-programming/input-output-import>
- [8] (2021) w3schools.com. [Online].
https://www.w3schools.com/python/python_datatypes.asp