



---

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

## II. Methods

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

- **Variables and Data Types**

In Python, there are names/codes that 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 data type of a variable. 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:

<b>Text Type</b>	<code>str</code>
<b>Numeric Types</b>	<code>int</code> , <code>float</code> , <code>complex</code>
<b>Sequence Types</b>	<code>list</code> , <code>tuple</code> , <code>range</code>
<b>Mapping Type</b>	<code>dict</code>
<b>Set Types</b>	<code>set</code> , <code>frozenset</code>
<b>Boolean Type</b>	<code>bool</code>
<b>Binary Types</b>	<code>bytes</code> , <code>bytearray</code> , <code>memoryview</code>

- **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.

[3] Here are the assignment operators used in 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 = 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

Comparison operators are used to compare values. It returns either True or False according to the condition. [4] 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 is simply 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 includes 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 which is also the most common operator in Python is `print()` wherein the user can simply 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 in it. It also takes optional parameters start and end to specify the starting and ending positions in the string respectively. [6]

Looping statements are continuous running loops that is given a certain condition and if that condition was met, the loop will end. [1] First type of looping statements 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 a loop statement is for which executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. [7]

- **Flow Control**

Condition statements in Python are if, if-else, and elif statements. If statement is one of the most commonly used conditional statement in most of the 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 code 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, elif statement is used to check multiple conditions only if the given if condition 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. [8]

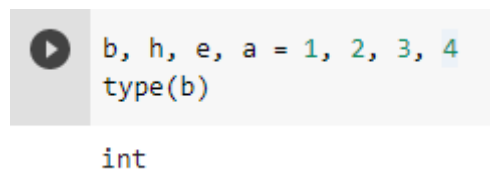
- **Functions**

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 requires parameters while non-parametric functions doesn't require any parameters, they're like void functions. [1]

The researcher also created a program wherein the objectives are met specifically: a program that computes the semestral grade of a student in two decimal places, be able to input and display their name and course, and display an emoji based on how high the student's semestral grade is.

### III. Results

#### A. Python Fundamentals



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

int
```

The image shows a code editor window with a play button icon on the left. The code inside consists of two lines: `b, h, e, a = 1, 2, 3, 4` and `type(b)`. Below the code, the output `int` is displayed. A horizontal line is positioned below the output.

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. Wherein, 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 mostly 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 the 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 variable i.

## B. Grade Calculator

Please refer to [this link](#) 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 the aim of this laboratory activity.

```
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 objectives of the program and that is to display an emoji as a feedback for how high the grades of the student has. In Line 6-8, the UniCode of the emojis are inserted as well as some basic expressions and a short message to be displayed as a feedback once the average of grades are computed. On

Line 10-15, the researcher used an elif statement so that if the average grade is above 70, it will display a happy emoji and if it's false, it will check the condition on the elif block which is equal to 70, and if both conditions are false, the program will then proceed to else block and display a sad emoji.

[illegible]

Figure 4.2 Input and Output Operations

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

## IV. Conclusion

Programming needs a lot of practice and effort for us to understand and be able to make our own program. Back when I was in 11<sup>th</sup> grade, I never thought that I will be facing programming for the rest of my life because I don't have any background on it except for the scripts I'm using to clean Twitter accounts (clean means to delete followings, tweets, likes, etc.) that I mostly found in github, that's why I'm very doubtful of myself when I entered college under the BS Computer Engineering course because I know that most of my blockmates know more than me but after one semester, I found out that it is very interesting to learn and explore. What I love the most in programming is when I troubleshoot codes, the satisfaction after being able to make the program work is priceless. From 1<sup>st</sup> year 1<sup>st</sup> semester until today, we have already learned 3 programming languages namely, C++, C#, and Python and I only found out today that it is really necessary to learn more than one language because I can use it after I graduate especially when I have to translate from Python to a more basic language like C++.

This laboratory activity helped me conclude that Python is one of the advanced programming languages and it is very complex but when you try and practice a lot, it can be understandable and enjoyable. I was also able to determine the specific data types and such, as well as determine what specific function a function can be used for. This laboratory activity widen my knowledge and it feels like I was able to learn Python from the beginning again but now with smaller steps like knowing what is a variable first and I think that's what I really need because I will be with this programming language more often. Moreover, I was also able to learned that Python can display emojis just by inserting Unicodes (and there are two more ways based on what I've read) and was also able to discover that I can use special characters as a border too. This laboratory activity made me want to learn and explore more.

## References

- [1] D.J.D. Lopez. "Python Fundamentals" [Online]. Available: [https://drive.google.com/a/uci.edu/file/d/0BwBjTzNOHOvCV1owMGlzWXB3QTA/view?usp=drive\\_web](https://drive.google.com/a/uci.edu/file/d/0BwBjTzNOHOvCV1owMGlzWXB3QTA/view?usp=drive_web)
- [2] "python\_datatypes @ www.w3schools.com." [Online]. Available: [https://www.w3schools.com/python/python\\_datatypes.asp](https://www.w3schools.com/python/python_datatypes.asp).
- [3] "gloss\_python\_arithmetic\_operators @ www.w3schools.com." [Online]. Available: [https://www.w3schools.com/python/gloss\\_python\\_arithmetic\\_operators.asp](https://www.w3schools.com/python/gloss_python_arithmetic_operators.asp).
- [4] "operators @ www.programiz.com." [Online]. Available: <https://www.programiz.com/python-programming/operators>.
- [5] "input-output-import @ www.programiz.com." [Online]. Available: <https://www.programiz.com/python-programming/input-output-import>.
- [6] "count @ www.programiz.com." [Online]. Available: <https://www.programiz.com/python-programming/methods/string/count>.
- [7] "python\_for\_loops @ www.w3schools.com." [Online]. Available: [https://www.w3schools.com/python/python\\_for\\_loops.asp](https://www.w3schools.com/python/python_for_loops.asp).
- [8] "664153eb1eb690efe8a46de43c914d364c4f7e57 @ www.softwaretestinghelp.com." [Online]. Available: <https://www.softwaretestinghelp.com/python/python-conditional-statements/>.