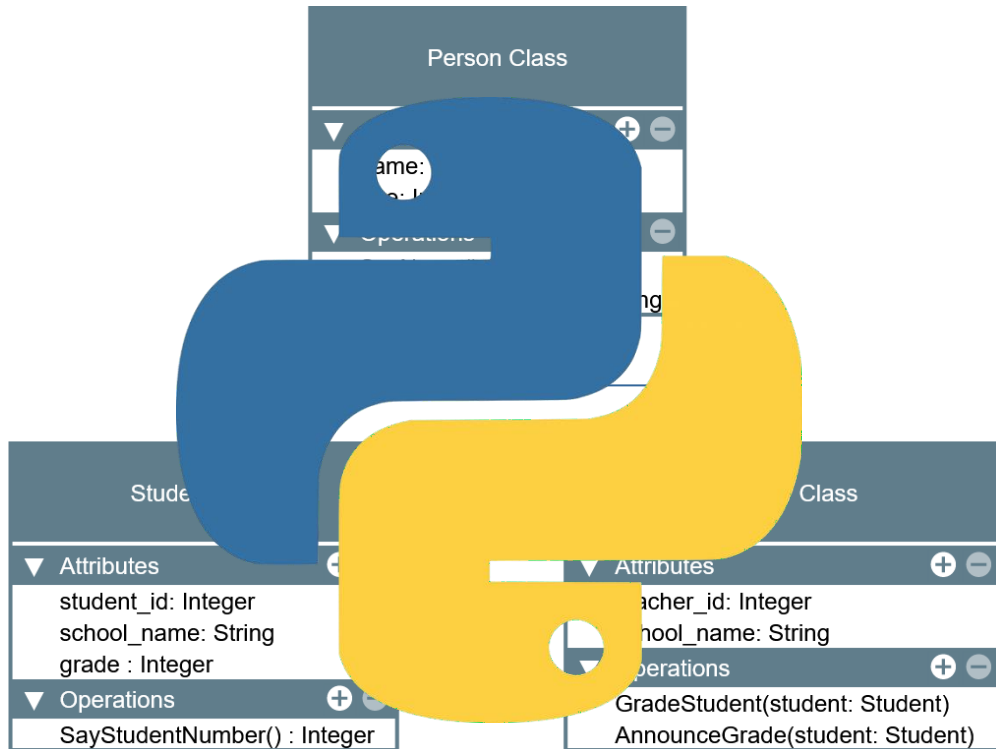




UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025



LABORATORY MANUAL

Object-Oriented Programming (CPE 103)



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

Laboratory Activity No. 2.1	
Literals, Operators, and Variables	
Course Code: CPE103	Program: BSCPE
Course Title: Object-Oriented Programming	Date Performed: 1/25/2025
Section: 1A	Date Submitted: 1/25/2025
Name: Ampong, J-kevin L.	Instructor: Maam: Sayo
1. Objective(s):	
This activity aims to familiarize students in the various data types of Python, assign values to variables, and perform operations in a Python program.	
2. Intended Learning Outcomes (ILOs):	
The students should be able to: 2.1 Assign different values to variables in Python 2.2 Perform different operations available with variables in Python	
3. Discussion:	
<p>The Python programming language is an interpreted language meaning the lines are evaluated line -by-line at runtime because there is no compile time at Python. This means that Python can dynamically allocate memory to variables as needed depending on the line of code that it interprets that is why Python is also referred to as a Dynamically typed language.</p> <p>Like other programming languages such as C/C++ and Java, Python can also assign values to specific blocks of memory through variables as well as perform operations such as but not limited to Addition, Subtraction, Multiplication, Division, and Modulo(remainder). This activity will focus on assigning values and performing operations in Python.</p> <p>Recall that a variable is a name that points to a specific location in memory where the data is stored. A variable can be allocated memory based on the data type it is assigned with which in Python can be: Integer, Float, Complex Number, Boolean, and String. In Python, lists, tuples, and dictionaries are also referred to as data types specifically sequences. More information can be found here (https://docs.python.org/3.8/reference/datamodel.html?highlight=data%20type#objects-values-and-types). These will be discussed further in lab activities.</p> <p>Variables in Python are assigned in the following manner:</p> <div style="text-align: center;"><code>variable_name = value</code></div> <p>Literals refers to the raw data given in a variable or constant. Literals can be some of the following: Numeric, Complex, String, Boolean, Special. Other literals are list, tuple, dict, set, and Unicode literals.</p>	
4. Materials and Equipment:	
Desktop Computer with Anaconda Python /Python Colab Windows Operating System	



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

5. Procedure:

Perform the activity using the Jupyter Notebook

This activity can be done either locally on Anaconda's Jupyter Notebook or online through Google Collaboratory which offers a free Jupyter Notebook environment for Google Users. IPython Notebook files (.ipynb) that are saved in the Google Drive can be opened on Google Collaboratory. Additional guides are available on the IPython Notebook template file that is provided with this activity. If the template is not present, these are the valuable links for reference:



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

Assigning variables of different data types in Python

1. In an empty cell, declare a variable **value** and assign it the value of 5 then display its value using `print()`. Please refer to this link:
2. Create a new cell and type the command: `type(value)` then run the cell. The output should be like the image below.

```
In [3]: type(value)
Out[3]: int
```

3. In a new cell, use the same variable **value** and assign it the value of 5.0 then print the value.
4. Repeat step 2.
Note: You may choose to decide how you execute the code in the cells for the next tasks in the procedure.
5. Repeat these steps for the following values:
 - a. `2+3j`
 - b. `'Hello World'`
 - c. `"Hello World"`
 - d. `True`
 - e. `False`
 - f. `[1,2,3,4,5]`
 - g. `(1,2,3,4,5)`
 - h. `{ 'name': 'Your_name' }`
 - i. `None`
6. Re-assign the **value** variable to be equal to 5.
7. Declare a new variable named **value2** to be equal to -6.

Please refer to this link:

https://colab.research.google.com/drive/1YVEaJWtYbiUDML_v29Ou9DDgp0cpX3Up#scrollTo=sb3ANVawxT7j



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

Performing Operations with Python

1. Using **value** and **value2**. Type the command: `print(value+value2)`
2. Repeat step 1 for the following values of **value** and **value2**:
Hint: You may try using this assignment **value, value2 = 5, -6** in the Notebook for the following steps:
 - a. value, value2 = 5.0, 6
 - b. value, value2 = -5, 6.1
 - c. value, value2 = "Hello", 'world'
Note: Modify the code so that hello and world would be separated.
 - d. value, value2 = [1,2,3], [4,5,6]
 - e. value, value2 = (1,2,3), (4,5,6)
 - f. value, value2 = {"name":"Royce"}, {"age":2}
Note: Observe the outputs carefully and try repeating them using subtraction.



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

3. Using value, value2 = 30, 4. Type the commands:

- a. `print(value*value2)`
- b. `print(value2**2)`
- c. `print(value2**3)`
- d. `print(value*value2+value2**2+1)`
- e. `print(value/value2)`
- f. `print(value%value2)`

Receiving Input Data using Python

Data can be received through keyboard input in Python by using the `input()` function. The input function has the following syntax:

`input("Message Name")`

The "Message Name" is an optional String parameter that can be customized to prompt the user for a message instead of having to print a message prompt separately. The default return value of the `input()` function is a String containing the value received from the keyboard. This value can be assigned to a variable shown in the example below:

`name = input("Enter your name: ")`

Please refer to this link:

https://colab.research.google.com/drive/1YVEaJWtYbiUDML_v29Ou9DDgp0cpX3Up#scrollTo=aHTm6HcfrZ57&line=1&uniqifier=1



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

Assigning Input Data to a Variable

Finding a person's BMI (metric)

1. Declare a new variable named **name** and assign it the value `input("Enter your name")`
2. Create another variable named **weight** and assign it the value `input("Enter your weight(kg): ")`
3. Create another variable named **height** and assign it the value `input("Enter your meters(m): ")`
4. Declare another variable called **bmi** and assign it the formula $bmi = \frac{weight}{height^2}$
5. Address the errors displayed step#4. You can accomplish this by converting the String input to another data type. An example would be:

```
weight = input("Enter your weight(kg)")  
weight = float(weight)
```

Or simply **weight** = float(input("Enter your weight(kg): "))

There are many functions available that can convert one data type to another. Some of which are the following:
`int()`, `float()`, `str()`

Other functions which maybe used in the later lab activities are: `complex(real, imaginary)`, `list()`, `tuple()`, `set()`, `dict()`, `ord()`, `bin()`, `hex()`, `oct()`.

6. Print the persons's name, weight, height, and bmi
Name: John Ray
Weight: 60
Height: 1.6764
BMI = 21.3499

Guide: 5.5 feet ~ 1.6764 m



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

Hint: You can combine two values by converting the output value to String and Concatenating (Addition) the operator on two strings.

```
print("Value: "+str(12))
```

You may explore many other methods to format values onto the print() function in Python. Another example is the following:

```
print("Value: ", 12)
```

please refer to this link:

https://colab.research.google.com/drive/1YVEaJWtYbiUDML_v29Ou9DDgp0cpX3Up#scrollTo=fgingjMLxT7n



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

6. Supplementary Activity:



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

Tasks

Please refer to this link:

https://colab.research.google.com/drive/1YVEaJWtYbiUDML_v29Ou9DDgp0cpX3Up#scrollTo=dDD41pzVAtOd&line=1&uniqifier=1

1. Write the Python equivalent code of the following C code:

```
int main(){
    float base = 0, height = 0, area = 0;
    printf("Enter the base of the triangle: ");
    scanf("%f", &base);
    printf("Enter the height of the triangle: ");
    scanf("%f", &height);
    area = (1/2)*base*height;
    printf("The area of the triangle is %f", area);
}
```

2. Write a program that would convert Celsius to Fahrenheit given the formula: $F = (C \times 9/5) + 32$
Example of conversion:

$$0^{\circ}\text{C} = 32^{\circ}\text{F}$$
$$-20^{\circ}\text{C} = -4^{\circ}\text{F}$$

3. Write a program that can determine the distance between two points given the coordinates using the formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Hint/Rule: No library or package is needed to implement this equation.

Example: $x_2, y_2 = -3, 3$ and $x_1, y_1 = 2, 2$ $d = 5.099019514$

Questions:

1. Give one major difference in syntax that Python has with other languages such as C?
In contrast to C, which utilizes curly brackets {} to define code blocks, Python uses indentation. This indicates that in Python, the whitespace at the start of a line is important and establishes the structure of the code.
2. How does variable assignment differ in Python compared with other languages such as C?
Because of Python's dynamic typing, you can give a value to a variable without first explicitly declaring its type. The = operator is all you need to do. C, on the other hand, needs type declaration, or static typing.
3. Try assigning variable names that start with numbers, and special characters. Is the assigning of variables that start with numbers accepted by Python? For Special Characters? Is there an exception for variables special characters?



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

Python variable names cannot begin with a number. Attempting to do so will result in a syntax error. The underscore `_` can be used inside names, but other special characters are usually prohibited. Variables like `__init__`, which have unique semantics in Python, are an exception.

4. Do the assignment operators (+, -, *, /, %, **) work for all data types? Why or Why not?

Assignment operators (+, -, *, /, %, **) often carry out mathematical operations, but the sorts of data they work with can affect how they behave. We refer to this as operator overloading. For instance, + adds numbers and concatenates texts and lists.

5. How does the * operator differ from the ** operator?

The operator * multiplies, for example, `5 * 3` to get 15. The operator, on the other hand, raises a number to a power through exponentiation. For example, `5 ** 3` is equal to 5 raised to the power of 3, or 125~.



UNIVERSITY OF CALOOCAN CITY
Caloocan, 1400 Metro Manila, Philippines

COLLEGE OF ENGINEERING
Computer Engineering
2nd Semester, School Year 2024-2025

6. Conclusion:

In summary, Python has a unique syntax emphasizing readability with indentation and dynamic typing. Variable naming has specific rules, with limitations on numbers and special characters. Operators exhibit flexibility due to overloading, and the * and ** operators perform multiplication and exponentiation, respectively. Understanding these core concepts is crucial for writing effective Python code within the Google Colab environment.

8. Assessment Rubric: