Laboratory Exercise
Understanding Algorithm with Python

Objective:

At the end of the exercise, the students should be able to:

- Construct a Python script based on the given algorithms.

Software Requirement:

- Python 3.7 or higher

Procedure:

1. Study the following sample Python syntaxes:

Task	Sample Syntax	Remarks
Variable declaration	`x = 5`	No need to indicate the data type. Do not add semi-colon.
Comment	`#This is a comment.`	
User input	`name = input("Enter your name")`	
Convert string to `int` or `float`.	`int(x)` `float(x)`	Convert to compare or compute values.
Convert `int` or `float` to `string`.	`str(x)`	Convert if the output must show a number and a string.

Combine conditional statements.	`x > 5 and x <10`
	`x > 5 or x < 10`
	`not(x < 5)`	

`if` statement	`if x == 0:` `    print("Enter a higher number")`	Use four (4) spaces per indentation level.
`else-if` statement	`elif x == 0:` `    print("Enter a higher number")`	
`else` statement	`else:` `    print("Try again")`	
Create a list of items.	`my_list = ["red", "blue", "green"]`	Python recommends the use of lists than arrays.
Display all items in a list.	`print(my_list)`	
Create an empty list.	`my_list = []`	
Add an item to the end of a list.	`my_list.append("yellow")`	

2. Open **Notepad++**.

3. Create a Python script that will compare two (2) numbers entered by the user. Refer to the sample syntaxes and the algorithm below.

   3.1. User enters the first number.

   3.2. User enters the second number.

   3.3. If the first number is less than, greater than, or equal the second number, a message is displayed.

4. Save the script as **algo1.py**.

5. To test and run the script, open Command Prompt. Navigate to your file's location then type **python algo1.py**.

6. Create another Python script that will display the names of your three (3) classmates. Refer to the sample syntaxes and algorithm below.

   6.1. User enters the name of the first classmate.

   6.2. User enters the name of the second classmate.

   6.3. User enters the name of the third classmate.

   6.4. The name of the three (3) classmates are displayed.

7. Save the script as **algo2.py**.

8. Create a folder named *LastName_FirstName_MI_LE1* (ex. Reyes_Nika_P_LE1) in your local drive.

9. Move the two (2) scripts to your folder.

GRADING RUBRIC (100 points):

Criterion	Description	Max Points
Correctness	The code produces the expected result.	40
Logic	The code meets the specifications of the problem.	40
Efficiency	The code is concise without sacrificing correctness and logic.	10
Syntax	The code adheres to the rules of the programming language.	10
TOTAL		**100**