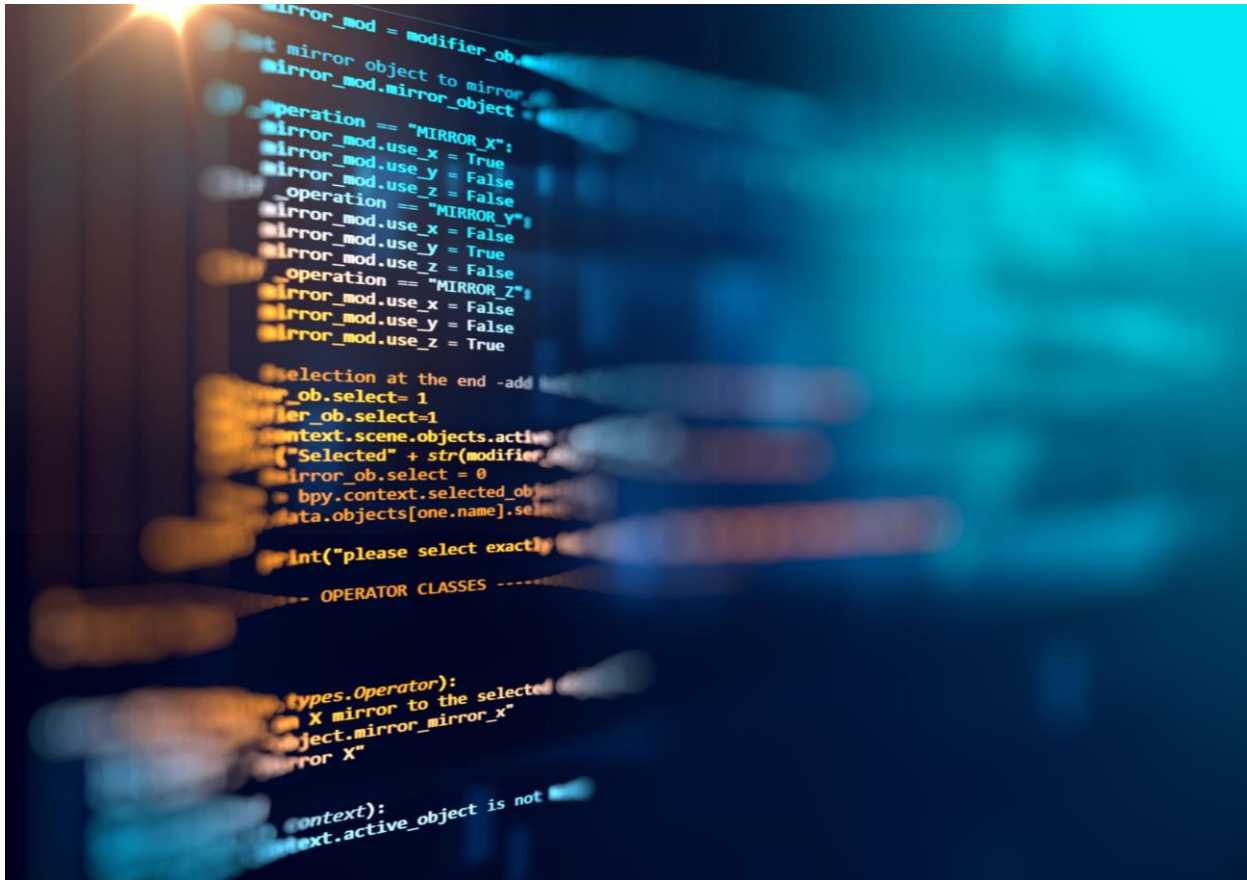


LBYCPA1

Programming Logic and Design Laboratory



Laboratory Module 3

Booleans, Control Structure - Conditional Statements

By

John Carlo Theo S. Dela Cruz | LBYCPA1 | EQ1

INTRODUCTION

Module 3 of this course is mainly focus about Booleans, Control Structure and Conditional Statements. Since we are developing our skills in programming. The module would provide us knowledge and understanding towards the flow control of programming. This part of the module for me is important since the flow control and conditional statements are always present in every scenario. The logical testing between true or false Boolean values. In this module, we will be focusing ourselves how can we manipulate and how to process these required programs that would be done in the Jupyter notebook software.

The main objective of every laboratory report is not to contribute to the field of knowledge, but to provide the opportunity for learning, and to provide further experience in the field of programming despite the hindrance of the Pandemic. Through the process of programming along side with the method of planning which is the use of Algorithms, Pseudocodes and flowcharts provides the step-by-step thought process of how the program works. Python is very flexible, and we are responsible on how we can approach every problem. There would be many syntax and logical errors along the way, but each programmer will pass through those challenges until we meet the set standards. With the help of Jupyter Notebook, w3school, and Stack Overflow, it provides us the right syntax on how we execute the code. This part of the module would be a game changer since it will deal a lot of conditions and more about the flow control of the program.

What do you think are the main objectives for this module? (Enumerate as many as you can.)

(a) Objectives

1. To familiarize with the Boolean data type
2. To understand the importance of flow control
3. To use comparison and Boolean operators in conditional statements
4. To solve computational problems using conditional statements.
5. To implement and apply flow control
6. To understand flow control

What are the materials used for this module?

(b) Materials and Tools

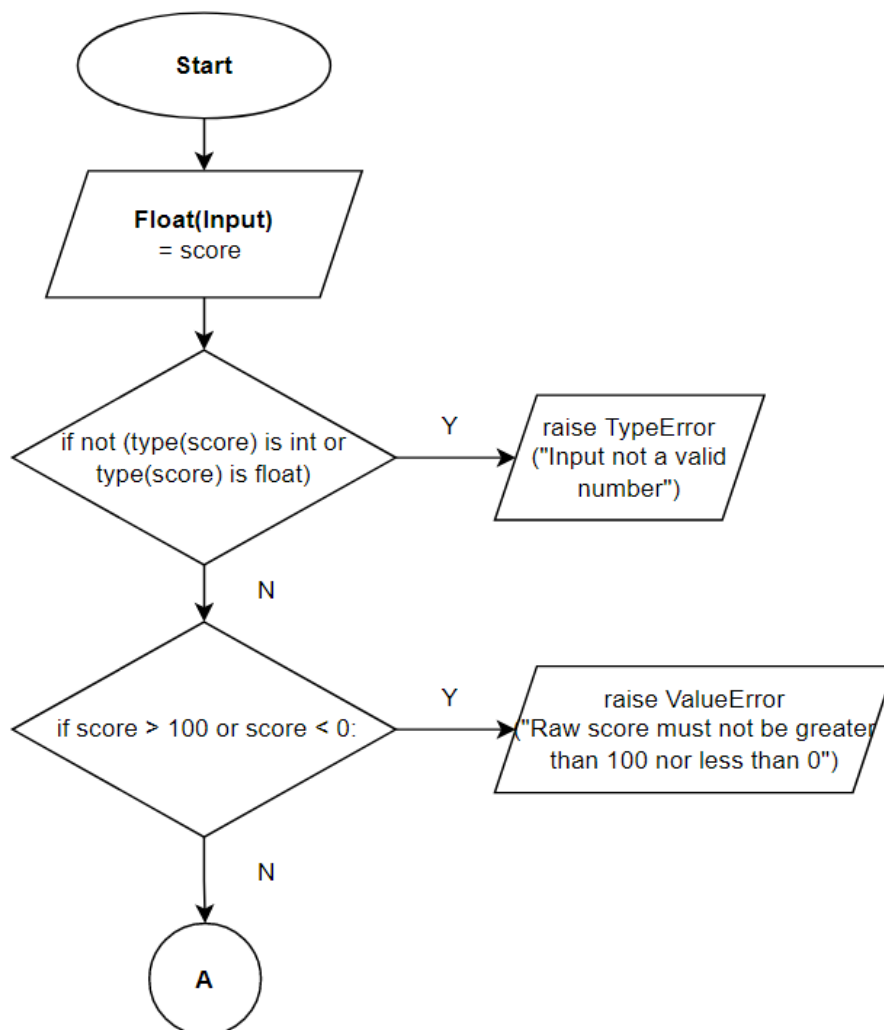
1. Module 3
2. Engineering Notebook
3. Diagrams.net
4. Jupyter Notebook
5. Google Browser

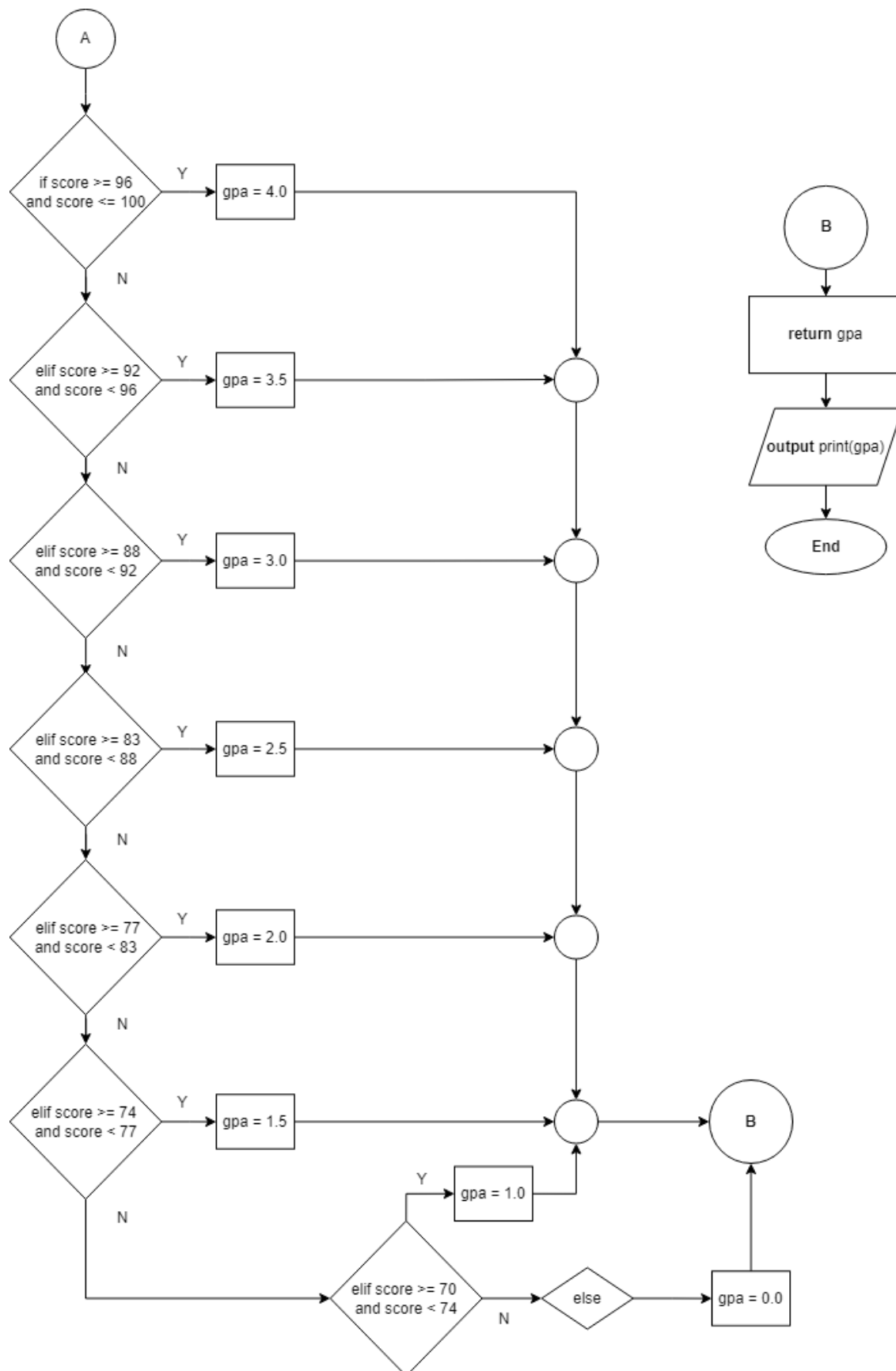
PROCEDURES (*Individual*) / EXPERIMENTAL PLAN

In every experimental plan of every Laboratory Report we always place our initial plans before performing executing our codes, in programming we always use Pseudocodes / Algorithms or Flowcharts, as a representation of how we can show the separate steps of each process in a sequential manner.

1. Familiarization Exercise 1 GPA Calculator:

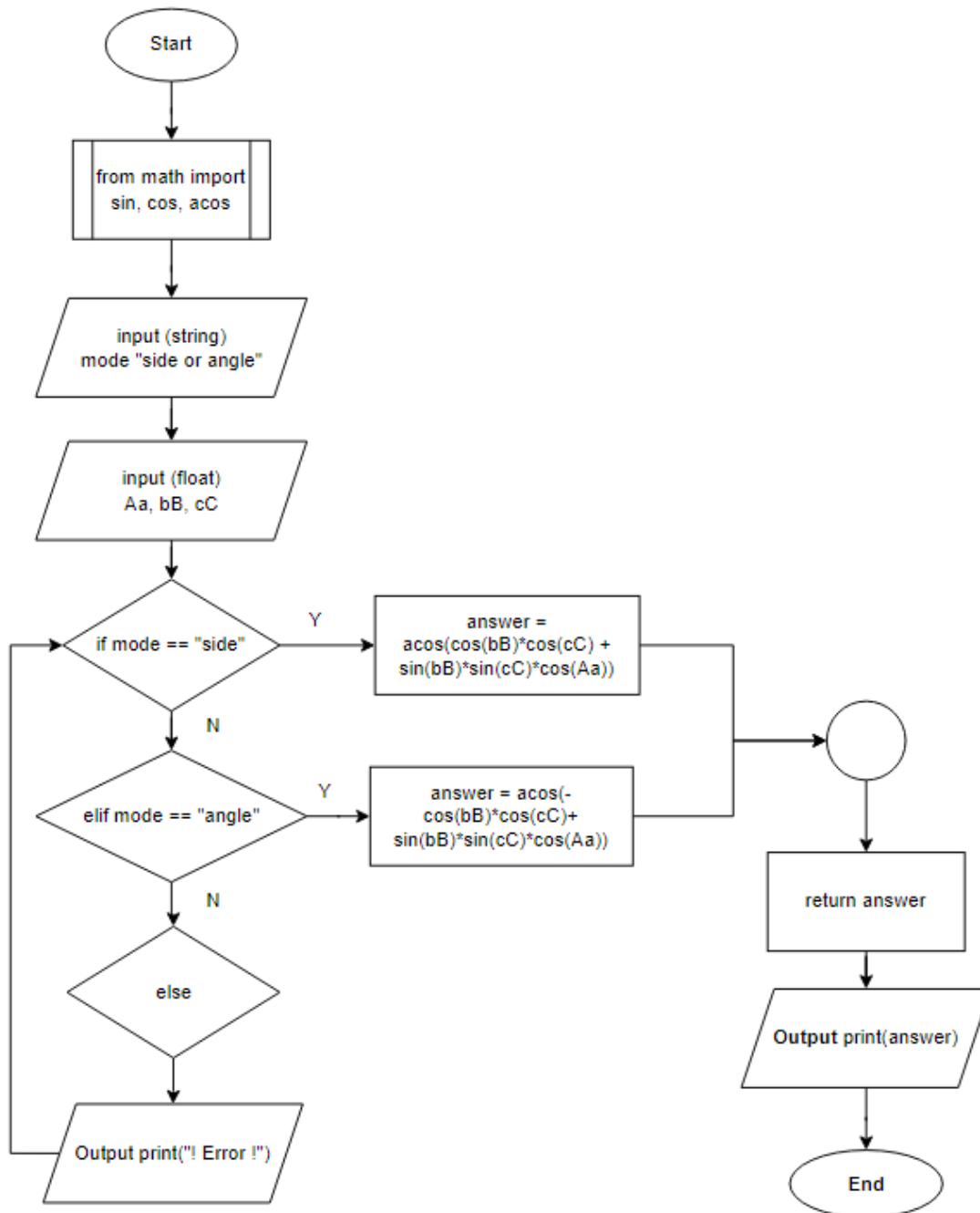
In this program of GPA Calculator, the program requires the user to input their raw grade which is between the value of 0 and 100. Once they inputted their grade, the program will verify if their input is an integer or a float and will verify if it is inside the set value of 0 and 100. After verifying the input, the program will find a condition on where the value will be satisfied according to the user's input. Once the program identifies the GPA, according to the user's input, the program will now then print the GPA.





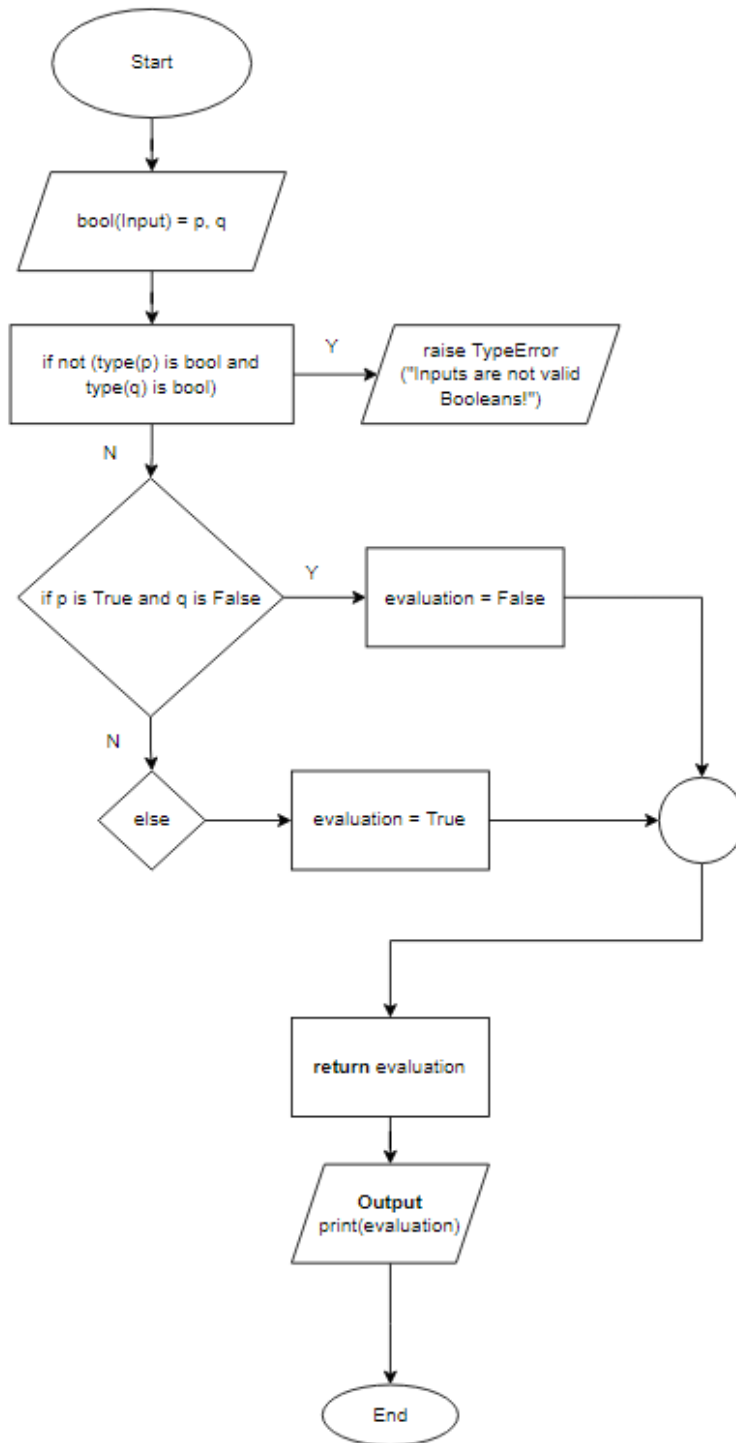
2. Familiarization Exercise 2 Spherical Trigonometry Calculator:

This program is a Spherical Trigonometry Calculator which is like the previous module in which we will be using mathematical equations to perform and execute the program. This program requires the users to input their preferred, its either side or angle. After inputting the preferred mode, the program requires the user to input 3 sides or values to be solved by the program. After solving, the program will print out the final answer.



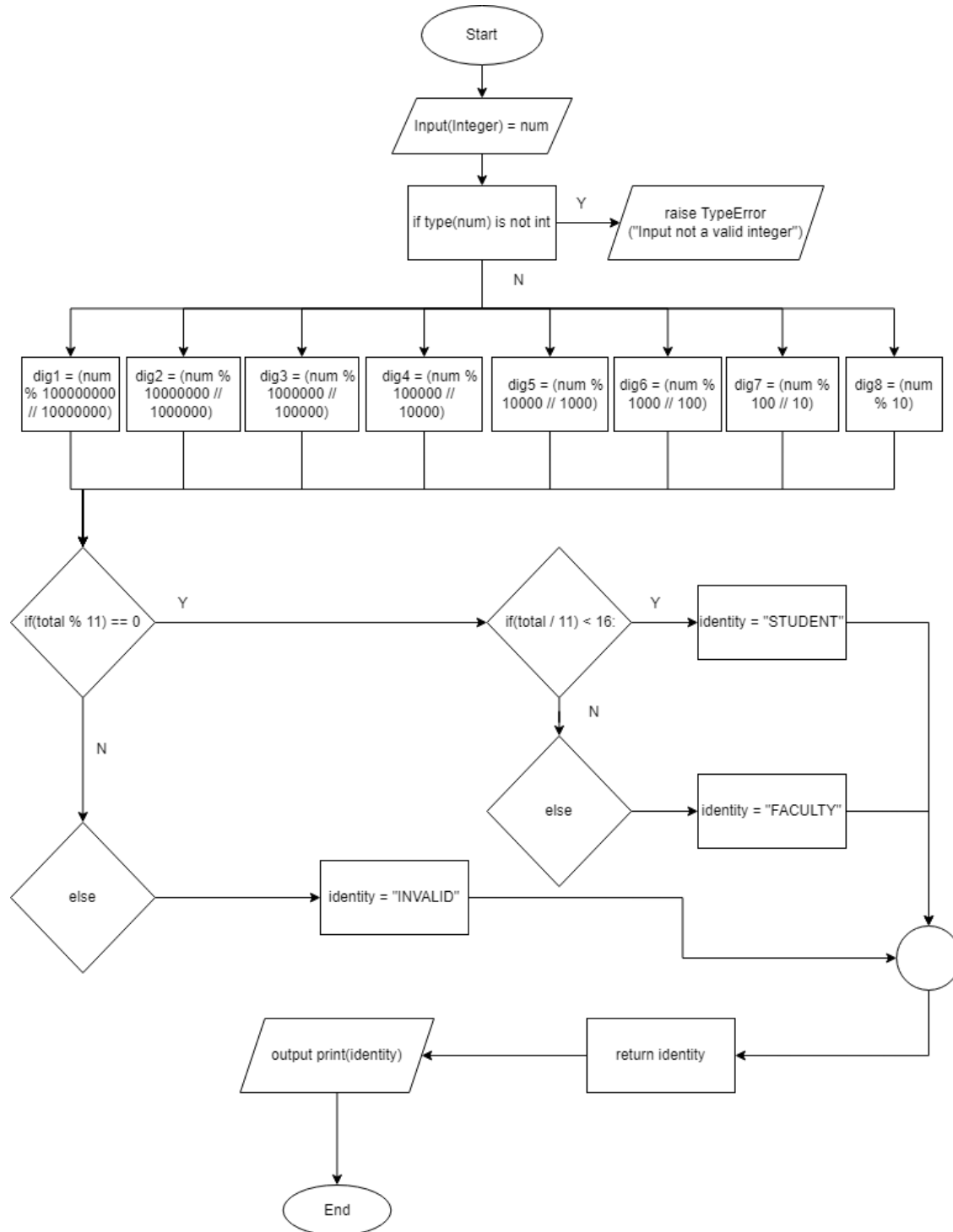
3. Familiarization Exercise 3 Conditional - Implication:

This program is a conditional logical operation in which it uses Boolean operators and conditional statements. The input would be 1 if True and 0 if False, and the program would evaluate whether the combination will be either true or false, in this case of conditionals only the combination of 1,0 will always be false.



4. Familiarization Exercise 4 ID Number Identifier:

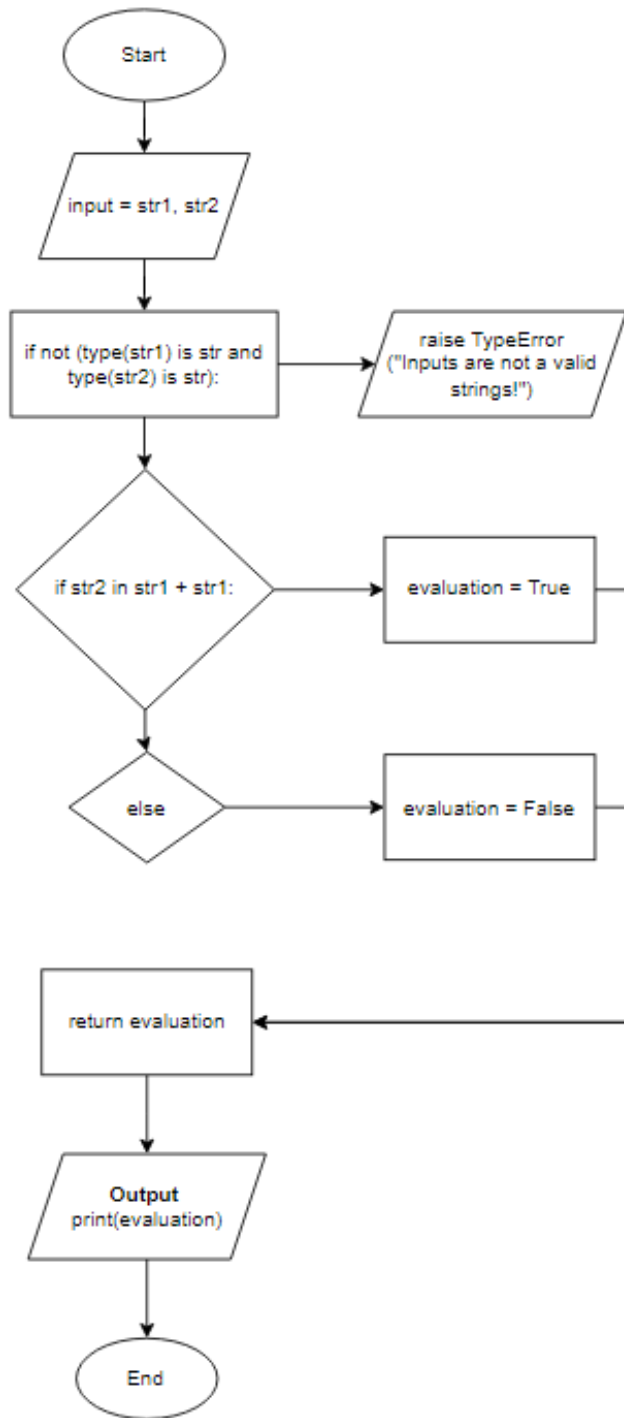
In this program we will use the knowledge and thought process in the last module, in which we will get the value of each digit individually and the program will multiply it by a specific number and will get the total. The program requires the users to input their ID, and the program will identify whether you're classified as a Student, Faculty or your ID Number is Invalid.



5. Familiarization Exercise 5 String Rotation:

This program is a string rotation in which the program would need the user's input to string statements and the program verifies and checks if both string statements are reversible. Then it prints the result whether it is considered as a String Rotation or not. (True or False)

Ex: (AB, BA == True)



CRITERIA	EXEMPLARY (90-100)	SATISFACTOR Y (80-89)	DEVELOPING (70-79)	BEGINNING (below 70)	WEIGHT
Experimental Plan (Flowchart/ Algorithm) <i>(SO-PI: B1)</i>	Experimental plan has supporting details and diagram/algorithm that is stated and well explained	Experimental plan has supporting details and diagram/algorithm that is stated but not explained	Experimental plan is vague or brief. It has supporting details and doesn't have diagram/algorithm	No experimental plan presented	30%
Conducting Experiment/ Activity <i>(SO-PI: B1)</i> <i>(SO-PI: K1)</i>	Objective and Materials used (modern engineering tools, software, and instruments/equipment) are identified. Steps are easy to follow for conducting and experiment/activity.	Objective is not stated. Materials used (modern engineering tools, software, and instruments/equipment) are identified. Steps are easy to follow for conducting and experiment/activity	Does not provide enough information to conduct an experiment/activity	No Objective, Materials used (modern engineering tools, software, and instruments/equipment), and steps for conducting experiment/activity provided.	20%

RESULTS AND DISCUSSION/COMPUTATIONS *(Include the program output screenshots, and discussions per problem solution)*

1. Familiarization Exercise 1 GPA Calculator:

Explanation: This program is a raw grade to DLSU GPA Calculator or Converter, the required input would be the raw grade of the student which has a condition, in which the user should only input values between 0 to 100. Before the process of getting the GPA, the program verifies the user's input whether it is an integer (whole number) or a float (whole number that has a decimal point), once it is verified the program will also verify if the user's input is within the range or within the set value of between 0 and 100, after all verifications are done, Once the raw grade has been verified the program by the set conditions, the program will now identify the GPA according to the value of the raw grade:

Raw Score	Grade Equivalent
100% down to 96%	4.0
Less than 96% down to 92%	3.5
Less than 92% down to 88%	3.0
Less than 88% down to 83%	2.5
Less than 83% down to 77%	2.0
Less than 77% down to 74%	1.5
Less than 74% down to 70%	1.0
Less than 70%	0.0

Screenshot of the Result:

```

return gpa # Make sure to assign the result to the gpa variable

score = float(input("DLSU GPA Raw Grade Grade Calculator: "))
gpa = raw2grade(score)
print("GPA:", gpa)

```

DLSU GPA Raw Grade Grade Calculator: 90
GPA: 3.0

2. Familiarization Exercise 2 Spherical Trigonometry Calculator:

Explanation: The program calculates the Spherical Triangle depending on its chosen mode, whether it's the side or angle. Before executing the codes, we declare and imported the function Math and get the exact values of cosine, sine and inverse of cosine function. This The program is required to input what mode will the user prefers; (Side or Angle) In this problem the mode=side is considered to be the default mode of the program. After inputting the preferred mode, the program will now then use the formula according to the conditional statements stated: If the mode side is chosen, it will follow a specific formula on how to get the value of side; If (elif) the mode angle is chosen or define, it will follow a specific formula on how to get the value of the angle of the Spherical Triangle.

```

if mode == "side" or mode == "SIDE":
    answer = acos(cos(bB)*cos(cC) + sin(bB)*sin(cC)*cos(Aa))
elif mode == "angle" or mode == "ANGLE":
    answer = acos(-cos(bB)*cos(cC)+ sin(bB)*sin(cC)*cos(Aa))
else:
    print("! Error !")

```

Screenshot of the Result:

```
return answer # Make sure to assign the result to the answer variable
mode = str(input("Hello \nSide or Angle?: "))
Aa = float(input("Input Value A: "))
bB = float(input("Input Value B: "))
cC = float(input("Input Value C: "))

answer = trig_compute(bB, cC, Aa, mode="side")
print(answer)
```

```
Hello
Side or Angle?: side
Input Value A: 1
Input Value B: 2
Input Value C: 2
0.9021658790144664
```

3. Familiarization Exercise 3 Conditional - Implication:

Explanation: In this program, this tests the Boolean operator “and”. The best way to represent this is by the use of the truth table, the user would input two conditions (p and q) but in this case we’ll use the value of 1 and 0 since those 2 characters are the only things that the computer can understand (binary value). Since the ruling in Boolean every string is considered as True; Boolean is False when the value is 0, or it does not have any value in it. Based on the Truth table, the combination of 1,0 or True, False is always False; While every Combination is always True. In that case the Program sets a condition in which when the input is True – False (1,0) the evaluation is = False; else everything is = True.

IMPLICATION TRUTH TABLE		
P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

© chillmath.com

Screenshot of the Result:

```

    return evaluation # Make sure to assign the result to the evaluation variable

print("1 = True, 0 = False\n\n")

p = bool(int(input("Input P: ")))
q = bool(int(input("Input Q: ")))
evaluation = conditional(p,q)
print("The statement is : ", evaluation)

```

1 = True, 0 = False

Input P: 1

Input Q: 0

The statement is : False

4. Familiarization Exercise 4 ID Number Identifier:

Explanation: This program requires us to get each place value and multiply them in an orderly manner, lastly getting the total of it. After getting the total value of the 8-valued ID Number, the program will verify if the total value is visible by 11, or else the program will print out INVALID, since the program sets a condition that it will always accept value visible by the value of 11. After verifying the value if it is visible by 11, it will now go to another condition in which it will check if the total value divided by 11, after dividing the program sets that, if the value is less than 16, then the identity of that ID Number is a STUDENT, and if its greater than 16, their identity is a FACULTY.

In getting the value of each digit, we'll do the same thing we did in the last module, particularly the 3-digit Reverser, because it uses the same mechanic in which we'll get the value of each place value using the arithmetic equations such as modulo and division integer.

```

dig1 = (num % 100000000 // 10000000)
dig2 = (num % 10000000 // 1000000)
dig3 = (num % 1000000 // 100000)
dig4 = (num % 100000 // 10000)
dig5 = (num % 10000 // 1000)
dig6 = (num % 1000 // 100)
dig7 = (num % 100 // 10)
dig8 = (num % 10)
total = (dig1 * 8) + (dig2 * 7) + (dig3 * 6) + (dig4 * 5) + (dig5 * 4) + (dig6 * 3) + (dig7 * 2) + (dig8 * 1)

```

12195758



12195758 % 100000000

= 12195758 // 100000000

= 1

$$\left[(1 \cdot 8) + (2 \cdot 7) + (1 \cdot 6) + (9 \cdot 5) + (5 \cdot 4) + (7 \cdot 3) + (5 \cdot 2) + (8 \cdot 1) \right]$$

Screenshot of the Result:

```
    return identity # Make sure to assign the result to the identity variable

num = int(input("Enter your ID Number: "))
identity = id_verify(num) #declaring the formulated function
print(identity)
```

Enter your ID Number: 12195758
STUDENT

5. Familiarization Exercise 5 String Rotation:

Explanation: This program requires the user to input two series of strings and the program would test and the program checks the second input string if it is IN the first string added to itself once or can be reversible or rotatable. But before the process is being executed the program verifies the user's input if they inputted a string or not, once it's verified then the program will now execute and proceed to the processing part of the program in which it will check both strings.

Screenshot of the Result:

```
    return evaluation # Make sure to assign the result to the evaluation variable

str1 = (input("Input a series of letters and/or numbers: "))
str2 = (input("Input 2nd Letter/Number: "))

evaluation = string_rotations(str1, str2)

print("Rotation: ", evaluation)
```

Input a series of letters and/or numbers: brrt
Input 2nd Letter/Number: trrb
Rotation: False

The Results and Discussions constitutes the data criteria in the Lab Report Evaluation Rubric:

Codes/Data/ Program	Data is well utilized in the program. Program codes are easy to read. Program output has no error. Questions are answered completely and correctly	Data is somewhat utilized in the program. Program code are easy to read. Program output has an output but logically incorrect. Some questions are answered completely and correctly	Data is not utilized in the program. It has a missing significant code/syntax in the program.	No program presented	30%
------------------------	--	---	---	----------------------	-----

CONCLUSION:

This 3rd Module of LBYCPA1 mainly focuses on Booleans, Control, and Conditional Statements/Structures as well as the usage of operators. We were introduced with syntax: *not*, *and*, *or* as a conditional statement; *if*, *else if*, *else*, *match*, and *case* as a Logical Operators. This module tested my logical thinking and somehow challenged me on how to approach each problem in the module, in which I was in a situation that I know how the process works but I do not know how to execute the code, and I overcame that challenge by reading the module and watching tutorials in google to further understand each syntax in the said Logical Operators.

Also, I discovered that I should be in “a state of coding”, like flow state but in terms of coding; so, a programmer can focus and execute their plans and codes well. Visualizing what you want and how you will reach that point given your inputs. Having a calm state when coding, because losing that kind of state would cause stress and can prolong your code time. Lastly, taking a break each problem whenever you are lost in a situation, because taking breaks can calm you down enough to be able to start over again (Mental Reset).

The most difficult and time consuming are exercise 3 and 5, because I tried to think a complicated way of executing the code. I learned in those 2 exercises is for me to compromise myself and to think simple in this exercise to find out the right syntax in a short line of codes. Boolean and concatenating are somehow tricky; As I encounter more errors, and it gets to the point that I understand their purpose and function.

My recommendations for those who would be trying this activity is that you need to have the right practice and familiarize yourself with the different operators. Be in that state of coding so you can focus more on how you execute the code. Think creative and think outside the box.

Grammar, logical presentation, and format (SO-PI: G1)	The report was grammatically correct, logically presented and used the required format.	The report had minimal grammatical errors and somewhat presented logically. The required format was used.	The report had a lot of grammatical errors and not logically presented; the required format was barely used.	The report had a lot of grammatical errors, was not logically presented and the required format was not used.	20%
---	---	---	--	---	-----

REFERENCES *(Enumerate references in APA format)*

Chilimath (n.d.). Retrieved from. <https://www.chilimath.com/lessons/introduction-to-number-theory/truth-tables-of-five-common-logical-connectives/>

w3school (n.d.). Retrieved from. https://www.w3schools.com/python/python_intro.asp

stackoverflow (n.d.). Retrieved from. <https://stackoverflow.com/>

APPENDIX *(Attach all the source codes here per problem category)*

1. Familiarization Exercise 1:

```
def raw2grade(score):
    # This function converts a raw score to DLSU grade equivalent (70 passing score)

    if not (type(score) is int or type(score) is float): #It verifies a correct input
        raise TypeError("Input not a valid number")

    if score > 100 or score < 0:
        raise ValueError("Raw score must not be greater than 100 nor less than 0")

    if score >= 96 and score <= 100:
        gpa = 4.0
    elif score >= 92 and score < 96:
        gpa = 3.5
    elif score >= 88 and score < 92:
        gpa = 3.0
    elif score >= 83 and score < 88:
        gpa = 2.5
    elif score >= 77 and score < 83:
        gpa = 2.0
    elif score >= 74 and score < 77:
```

```

        gpa = 1.5
    elif score >= 70 and score < 74:
        gpa = 1.0
    else:
        gpa = 0.0

    return gpa # Make sure to assign the result to the gpa variable

score = float(input("DLSU GPA Raw Grade Grade Calculator: "))
gpa = raw2grade(score)
print("GPA:", gpa)

```

2. Familiarization Exercise 2:

```

def trig_compute(bB, cC, Aa, mode="side"):
    # This function computes the side or the angle depending on a mode
    variable, e.g. mode can be "side" or "angle".

    from math import sin, cos, acos # We will use sin, cos, and inverse cos
    functions

    # YOUR CODE HERE
    if mode == "side" or mode == "SIDE":
        answer = acos(cos(bB)*cos(cC) + sin(bB)*sin(cC)*cos(Aa))
    elif mode == "angle" or mode == "ANGLE":
        answer = acos(-cos(bB)*cos(cC) + sin(bB)*sin(cC)*cos(Aa))
    else:
        print("! Error !")
    return answer # Make sure to assign the result to the answer variable
mode = str(input("Hello \nSide or Angle?: "))
Aa = float(input("Input Value A: "))
bB = float(input("Input Value B: "))
cC = float(input("Input Value C: "))

answer = trig_compute(bB, cC, Aa, mode="side")
print(answer)

```

3. Familiarization Exercise 3:

```

def conditional(p, q):
    # This function implements the conditional logical operation

    if not (type(p) is bool and type(q) is bool):
        raise TypeError("Inputs are not valid Booleans!")

    if p is True and q is False:
        evaluation = False
    else:
        evaluation = True

    return evaluation # Make sure to assign the result to the evaluation
    variable

print("1 = True, 0 = False\n\n")

p = bool(int(input("Input P: ")))
q = bool(int(input("Input Q: ")))

```



```
evaluation = conditional(p,q)
print("The statement is : ", evaluation)
```

4. Familiarization Exercise 4:

```
def id_verify(num):
    # This function returns the strings "INVALID", "STUDENT", and "FACULTY"
    # for each corresponding ID category

    if type(num) is not int:
        raise TypeError("Input not a valid integer")

    dig1 = (num % 100000000 // 10000000)
    dig2 = (num % 10000000 // 1000000)
    dig3 = (num % 1000000 // 100000)
    dig4 = (num % 100000 // 10000)
    dig5 = (num % 10000 // 1000)
    dig6 = (num % 1000 // 100)
    dig7 = (num % 100 // 10)
    dig8 = (num % 10)
    total = (dig1 * 8) + (dig2 * 7) + (dig3 * 6) + (dig4 * 5) + (dig5 * 4) +
    (dig6 * 3) + (dig7 * 2) + (dig8 * 1)

    if (total % 11) == 0:
        if (total / 11) < 16:
            identity = "STUDENT"
        else:
            identity = "FACULTY"
    else:
        identity = "INVALID"

    return identity # Make sure to assign the result to the identity variable

num = int(input("Enter your ID Number: "))
identity = id_verify(num) #declaring the formulated function
print(identity)
```

5. Familiarization Exercise 5:

```
def string_rotations(str1, str2):
    # This function checks if the given two strings are rotations of each
    # other
    # Hint: You can use the 'in' operator and string concatenation

    if not (type(str1) is str and type(str2) is str):
        raise TypeError("Inputs are not a valid strings!")

    # YOUR CODE HERE
    if str2 in str1 + str1:
        evaluation = True
    else:
        evaluation = False

    return evaluation # Make sure to assign the result to the evaluation
    variable

str1 = (input("Input a series of letters and/or numbers: "))
str2 = (input("Input 2nd Letter/Number: "))
```

```
evaluation = string_rotations(str1, str2)
print("Rotation: ", evaluation)
```