CMPSC 102
Discrete Structures
Fall 2019

Lab 4 Assignment (two week lab):
Truth Tables and Applications of Usage
Submit deliverables through your assignment GitHib repository bearing your (team) name. Place source code and Markdown documents in their respective `src/` and `writing/` directories.

Objectives

To enhance your command of the Python programming language by completing a program to complete truth tables and to calculate Boolean equations for making decisions based on Boolean inputs. This lab is designed to give practice in programming functions, lists, dictionaries, Boolean value, conditional statements and other building blocks of the Python programming language.

GitHub Starter Link for Groups

STOP! STOP!
Not everyone will be clicking this link at this time!
Only the team leader will be clicking the link to create the repository!!
`https://classroom.github.com/g/vvkqeF0q`

Creating your repository

We will use a group assignment functionality of GitHub Classroom for this assignment. For group assignments **only one person will be creating the team while the other team members will join that team.** Please form a team of **no more than three people** and select one person to create the repository.

The selected person of the team should go into the link to the lab in the assignment sheet. Copy this link and paste it into your web browser. Now, you should accept the laboratory assignment and create a new team with a unique and descriptive team name (under “Or Create a new team”).

Now the other members of the team can click on the assignment link and select their team from the list under “Join an Existing Team.” When other team members join their group in GitHub Classroom, a team is created in our GitHub organization. Every team member will be able to push and pull to their teams repository.

Please work in groups: Unless you provide the instructor with documentation of the extenuating circumstances that you are facing, not working in a team and not accepting the assignment means that you automatically receive a failing grade for it.

To push your changes, you can use the following commands to add a single file, you must be in the directory where the file is located (or add the path to the file in the command):

- `git commit <nameOfFile> -m ‘‘Your notes about commit here’’`
- `git push`

Alternatively, you can use the following commands to add multiple files from your repository:

- `git add -A`
- `git commit -m ‘‘Your notes about commit here’’`
- `git push`

Reading Assignment

Please read the covered chapters from the course book, consult the week's slides and your class notes. You can also find useful information in the Python community by performing online research. Please take some time to gain experience with using Markdown to complete your writing assessments. See *Mastering Markdown* `https://guides.github.com/features/mastering-markdown/` for more details about Markdown. Another good reference may be found at: `https://markdown-it.github.io/`.

Table 1: Table for Proof of: $A \vee (B \wedge C) \equiv \ (A \ \vee B) \wedge (A \vee C)$

A	B	C	$B \wedge C$	$A \vee B$	$A \vee C$	$A \vee (B \wedge C)$	$(A \vee B) \wedge (A \vee C)$
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Part 1: Prove that an existing truth table is correct

In this lab you will first design a Python3 program,

file: `src/myTruthCalculator.py`

that completes the truth table shown in Table 1 (below) using the input lists of Figure 1. Your program is to build-out each column to display the logic of the table's headers. You will need to decide (as a group) how your code will function, however, please be sure to write a separate function for this part of your code for the purpose of organization and debugging. After completing this table, a comparison of its columns will prove that $A \vee (B \wedge C) \equiv \ (A \ \vee B) \wedge (A \vee C)$.

```
# Note: lists for making all possible inputs of data.
a_list = [True, True, True, True, False, False, False, False]
b_list = [True, True, False, False, True, True, False, False]
c_list = [True, False, True, False, True, False, True, False]
```

Figure 1: You and your group are to apply these lists of Boolean values to prepare each column of Table 1.

Sample output of the table

The output from your code should produce a completed truth table using keyboard-type graphics as shown in Figure 2, below.

```
+------------------------------------+
|     X    |    Y    |   X AND Y     |
+------------------------------------+
|   True   | True    |   True        |
|   True   | False   |   False       |
|   False  | True    |   False       |
|   False  | False   |   False       |
+------------------------------------+
```

Figure 2: A sample outputted table using graphics

Part 2: An Open-Ended Lab to Design Your Own Truth Table by Processing Data for Decision-Making

In the second part of your lab, you are to create a new truth table program,

file: `src/decisionMaker.py`

to be applied to decision-making using your group's imagination. To create this table, you will extract `True` and `False` Boolean values from user inputs (or a loaded file) and then to create a logical equation to which these inputs are applied. There are three inputs in this case and will be used to make *some* type of decision.

For example, imagine that you and your group are creating a decision making program to help a bank decide whether to give a client a loan. The bank's client would enter `True` and `False` in response to a series of questions which will be used by logical equations to determine whether or not to grant the client a loan.

In the example of the bank loan, some of your questions could take the following themes. Be sure to introduce questions with both `True` and `False` type answers.

- I have good credit : (`True` or `False`)

- I am have a steady job: (`True` or `False`)
- I have been arrested: (`True` or `False`)
- I have a family with two or more dependents: (`True` or `False`)
- I do not have any collateral for a loan: (`True` or `False`)

Note: your truth table will be made up of `True` and `False` values which are subjected to at least three operators (i.e., `AND`, `OR`, `NOT`), as well as an `Equivalent` column which will be used to determine how to act on the decision based on the inputs. You and your group are to determine the questions (**up to ten**), as well as, the logical equations that will be used to provide the decision based on the results of the questions. You do not need to display the entire table which could quickly become very big. Instead, you may use the inputs to determine the decision in a single line of code.

Questions for Your Discussion Document

In your `writing/discussion.md` file, you are to respond to the below questions-in-blue.

1. Introduce your application area where a decision is to be made.
2. Introduce and justify the questions that can be answered by `True` or `False`. For each input in your data you should explain how the question contributes to a calculated decision.
3. Introduce and describe your Boolean equation(s) that you use to compute your calculated decision. Justify and motivate the use of this equation. How does the equation work?
4. Output: Show and justify the decision from the inputs of the questions. In your justification, please be sure to argue that your questions provided enough information to reach a decision.

Required Deliverables

Submit deliverables through your assignment GitHib repository bearing your name, as well as all names in your group for your group work. Place source code in `src/`, the data in `inputs/` and the Markdown file `writing/` directories.

1. **Group Work with a Team Leader**: You are to work in groups to complete this open-ended lab. Give your group a name. Each member is to work collaboratively with the other group members. The group will submit all materials (code and documents mentioned below) to the team repository, which will be graded by the instructor. You are to use Markdown to format your written documents. **Please be sure to add the names of each member of your group in all your submitted work (i.e., code and markdown documents).**

2. File: **src/myTruthCalculator.py**: Your completed and working python code to create the truth table of Table 1. In this code, you will use functions and sets (such as lists, dictionaries) to create and contain your results. Use your code written in class to help inspire you.

3. File: **src/decisionMaker.py**: Your Python source code to accept data, determine `True` or `False` Boolean inputs from a user to make a decision. Be sure to use no more than ten (10) questions to supply inputs for your decision making algorithm and its logical equations.

4. File: **writing/resultsAndDiscussion.md**: A Markdown file that responds to the above *questions-in-blue.*