

**CMPSC 102**  
**Discrete Structures**  
**Fall 2018**

**Lab 4 Assignment (two week lab):**  
**Truth Tables and Applications of Usage**

**Submit deliverables through your assignment GitHub repository bearing your (team) name. Place source code and Markdown documents in their respective src/, inputs/ 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/0Uwf8kH0>

## 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 four 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/>.

## Additional Materials

Please locate your `src/` directory for this lab where you will find an empty file into which you will add your code for the completion of this lab. Specifically, in this lab you are provided with the following files:

- An empty Python3 source code; `src/myTruthCalculator.py` and `application.py`
- An empty csv file for your data if you choose to load your data into your program.
- An empty document to be completed in Markdown to describe your groups application of a truth table, showing trial runs of a decision being made using your found or invented data; `writing/discussion.md`.

## Part 1: Prove that an existing truth table is correct

In this lab you will first design a Python program that completes the proof shown in Table 1. Each column of your work will correlate to Python code to compute the True or False nature of the result in each entry. Your code will have to use sequences such as `lists`, `dictionaries`, or `sets` to input the True and False values into the table. By running your program, each column using previously calculated results. It is suggested that the entries of each column are placed into a list and used to supply the next column's calculation. Check your class notes for example code to get you started. Be sure not to hard-code your True and False values in your code beyond the A and B columns. For example, the column  $(A \vee B) \wedge (B \vee C)$  column is to be generated from the calculations of the previous columns,  $(A \vee B)$  and  $(B \vee C)$ .

### Sample output of each column

Your output from your code using functions has to show each of the columns but does not have to be in a tabular format, unless you choose to do so. For instance, your output for a column of OR and AND calculations of True and False inputs might resemble the following.

```
--OR Column --
True OR True : True
True OR False : True
False OR True : True
False OR False : False

--AND Column --
True AND True : True
True AND False : False
False AND True : False
False AND False : False
```

## 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 to be applied to decision-making using your group's imagination. To create this table, you will extract True and False Boolean values from a data set and then create a logical equation to determine how a decision is to be made using those inputs. You will use Python code to conduct all tasks in this application.

Your 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.

### Introduction

There are inputs which must be considered before decisions can be made. For example, in deciding to invest money into the stock market, a market-agent might have a series of criteria to consider for each company before deciding to proceed with an investment. The market-agent follows a series of questions which can be answered by True or False responses. For example, some of these questions may look like the following.

- Do I have enough money to invest at this time?
- Have I invested in this company before?
- Is the company stock healthy for a previous time-frame?
- Is the company likely to continue along in next fiscal period with the same economic health?

- Are there pieces of pessimistic news available that could interrupt this stock value in some negative way?
- and others

Once the questions have been answered for each company, the Boolean values of each are fed into a truth table to determine whether the company is in good economic standing. This truth table is created by using Boolean equations, containing operators (i.e., AND, OR, NOT) to weigh the “advantages” and “disadvantages” of a company concerning its input values.) The table uses each of its columns to work toward a decision column in which the actual decision is made. After the equation has been applied, a decision can be made about the wisdom of the investment using the previous Boolean inputs. In other words, a truth table of all possible outcomes from each question already exists and the actual answers are used to look-up the determined decision in the table.

## **The Decision-Making Algorithm: a Boolean Equation**

### **What Application?**

You and your group are to spend some time to discuss the kind of application that you would like to apply to your decision-making project. You will start by determining a type of decision and then to determine the Boolean equation which will be used to determine the outcome of this decision. Spend some time to discuss what kinds of questions you will ask to obtain Boolean values to be used to lookup values in your table to make your decision. Please make informed decisions which can be justified by some type of real-world application or scenario.

#### **0.0.1 What Questions to Ask?**

Once you have chosen your application, you are to determine three questions to answer by True or False. These will be used to look up the decision in your table.

### **What Truth Table?**

Next, calculate your entire truth table of all possible combinations of Boolean inputs and outputs. Make sure that each column leads to the Boolean equation which provides the decision.

If you have only three variables, then you will have  $3^2 + 1$  rows since each variable can have one of two possible outcome states (i.e., True or False). In other words, your table will be ten rows deep: one row is reserved for header information while the other nine subsequent rows will contain the exhaustive combination of all three variables (in combinations) as they are looked-up in your group’s truth table.

### **What Equation Decision to Make?**

After you know the decision to make, invent the mechanism of your Boolean equation to make a decision. Think in terms of your original application and what an objective would be for a good

decision-making. Your Boolean inputs can be made up, or come from real world sources (please add a reference if this is the case). Either way, your collected data will be looked-up in your truth table, containing all possible outcomes for for any three Boolean inputs. Your data for this task may be contained in lists. Note, the following lists are the inputs for an exhaustive table of all possible combinations of inputs.

```
# 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]
```

### What Conclusions Were Made?

In your discussion document, `writing/discussion.md`, you will describe your decision(s) and how your equation was able to help make an informed decision from your supplied inputs. You are to process data and extract its True and False values using Python code to make your decision. How this is done, is left to you and your group to decide. For instance, you could hard-code these lists (or other data container) in the source code or read them from a text file.

Please have at least ten different sets of data to apply to the Boolean equation to describe how your decision-making Boolean equation processes. This equation is described next. All code is to be done using Python by completing the file: `src/application.py`. Your data might be stored in lists. Note, the data stored in lists below can be used to create an exhaustive combination of all possible permutations of True and False trios. In this case, you would be able to find your True and False values in the rows to find the decision to make.

## Questions for Your Discussion Document

In your `writing/discussion.md` file, you are to respond to these questions.

1. Introduce and describe questions that can be answered by True and False. For each input in your data you should explain what True or False value corresponds to in regards to your data.
2. Introduce and describe your ten data sets. How were these sets chosen by your group?
3. Introduce and describe your Boolean equation to compute your final decision. Justify and motivate the use of this equation. How does the equation work?
4. Output: What was the decision that followed from applying each of the ten data sets to the boolean equation? Discuss how to interpret this decision in terms of the initial Boolean values and the Boolean equation.

## Required Deliverables

Submit deliverables through your assignment GitHub 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. **Use 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 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. **src/myTruthCalculator.py:** Your completed and working python code to calculate each column Table 1. In this code, you will use functions and sets (such as lists, dictionaries) to contain your results. After the inputs have been entered, all subsequent columns are to be created from previously calculated values. The columns are to be outputted to the screen. Use your code written in class to help inspire you.
3. **src/application.py:** Your Python source code to accept data, determine True or False from this data and then apply these Boolean values to some decision making algorithm.
4. **inputs/applicationData.csv:** Your data that you used to create your Boolean values, if it is not included in the source code. Remember that your group's data can be made-up or could be collected from a real-world source.
5. **writing/discussion.md:** A Markdown file containing the discussion of your application (its scenario), the questions you asked of your data, the Boolean results to the questions, and the decision-making Boolean equation to which you subjected your three sets of Boolean values from the data. Also, please use a though-provoking argument to justify your equation. Be sure to discuss its mechanism in your Markdown file `writing/discussion.md`. Be sure to describe how it works and why it was chosen for your application. What was the motivation to use this particular equation?

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	0	0	0
0	0	1	0	0	1	0	0
0	1	0	0	1	0	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1