Karnatak Law Society's
GOGTE INSTITUTE OF TECHNOLOGY
Udyambag Belagavi -590008
Karnataka, India.

A Seminar Report on

IMPLEMENTATION OF FUNDAMENTALS OF LOGIC.

Submitted for the requirements of 4th semester B.E. in CSE

for **"Discrete Mathematical Structures and Graph Theory(18CSMAT41)"**

Submitted by

NAME	USN
1) Shivani V Banke	**2GI20CS140**
2) Shradha Mallikarjun Patil	**2GI20CS144**
3) Srushti B Mudennavar	**2GI20CS158**
4) Yash Herekar	**2GI20CS184**

Under the guidance of

Dr. Shreenivas R Kirsur

Asst. Prof., Dept. of MATHEMATICS

Academic Year 2021-2022 (Even semester)

Karnatak Law Society's
GOGTE INSTITUTE OF TECHNOLOGY
Udyambag Belagavi -590008
Karnataka, India.

Department of Computer Science and Engineering

Certificate

This is to certify that the Course Project work titled **"Implementation of Fundamentals of Logic"** carried out by **Student SHIVANI BANKE, SHRADHA MALLIKARJUN PATIL, SRUSHTI B MUDENNAVAR, YASH HEREKAR** bearing **USNs:2GI20CS140, 2GI20CS144, 2GI20CS158, 2GI20CS184** for **Discrete Mathematical Structures and Graph Theory (18CSMAT41) Open Elective** course is submitted in partial fulfilment of the requirements for 4th semester B.E. in **COMPUTER SCIENCE AND ENGINEERING,** Visvesvaraya Technological University, Belagavi. It is certified that all corrections/ suggestions indicated have been incorporated in the report. The course project report has been approved as it satisfies the academic requirements prescribed for the said degree.

Asst. Professor	Head Of Department
Prof. Shreenivas Kirsur	**Dr. Sudhir Jog**
Department of Maths	Department of Maths

Head of Department
Dr. V Rajpurohit
Department of CSE

Academic Year 2021 - 2022 (Even semester)

Marks allocation:

	Batch No. :					
1.	Seminar Title: Implementation of Fundamentals of Logic	Marks Range	USN			
			2GI20CS140	2GI20CS144	2GI20CS158	2GI20CS184
2.	Abstract(PO2)	0-2				
3.	Application of topic to the course (PO1,PO2)	0-3				
4.	Literature survey and its findings (PO2)	0-4				
5.	Methodology, results and conclusion (PO1,PO3,PO4,PO5)	0-6				
7 6	Report and Oral presentation skill (PO9,PO10)	0-5				
	Total	20				

*** 20 marks is converted to 10 marks for CGPA calculation.**

Table of contents

PROBLEM STATEMENT :

Given the truth values of propositions p&q

a) Construct the truth table for De Morgan's Laws & Law's of Negation of a conditional & Distributive Law.
b) Write a program with specified Input & output for p&q, then construct the truth table for Demorgan's Law's & Negation of conditional & Distributive Law.

ABSTRACT :

Implementation of compound logical proposition equivalences using python language.

What is logic ?

- A science that deals with the principles and criteria of validity of inference and demonstration
- In simple way a proper or reasonable way of thinking about something
- It helps us to understand reasoning behind issues which could be applied in other issues as well

TOOLS USED :

- Vs code IDE
- Programming Languages – Python
- Module – ttg (truth table generator)

METHODOLOGY :

Laws of logic :

- Laws of logic are the Basic laws of Propositional Logic
- Fundamentally there are three laws of logic

1) the law of contradiction : not(a and not a)
2) the principle of identity : a is a
3) the law of excluded middle : either “a” or not “a”

Applications of laws of logic :

- Translating English Sentences into logical statements
- Logical Puzzles : They can be used for brain exercises, recreational purposes, and for testing a person’s reasoning capabilities
- Logic/Computer Circuits
- Artificial Intelligence – Fuzzy Logic : in fuzzy logic, there is an intermediate value too present which is partially true and partially false.
- Inference and Decision Making
- System Specifications

Laws of logic:

Equivalence	*Name*
$p \wedge \mathbf{T} \equiv p$ $p \vee \mathbf{F} \equiv p$	Identity laws
$p \vee \mathbf{T} \equiv \mathbf{T}$ $p \wedge \mathbf{F} \equiv \mathbf{F}$	Domination laws
$p \vee p \equiv p$ $p \wedge p \equiv p$	Idempotent laws
$\neg(\neg p) \equiv p$	Double negation law
$p \vee q \equiv q \vee p$ $p \wedge q \equiv q \wedge p$	Commutative laws
$(p \vee q) \vee r \equiv p \vee (q \vee r)$ $(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$	Associative laws
$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$ $p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$	Distributive laws
$\neg(p \wedge q) \equiv \neg p \vee \neg q$ $\neg(p \vee q) \equiv \neg p \wedge \neg q$	De Morgan’s laws
$p \vee (p \wedge q) \equiv p$ $p \wedge (p \vee q) \equiv p$	Absorption laws
$p \vee \neg p \equiv \mathbf{T}$ $p \wedge \neg p \equiv \mathbf{F}$	Negation laws

SOURCE CODE :

```
import ttg
def main():
    ch = 'y'
    while(ch == 'y'):
        print("1)~(p/\q)\t 2)~(p\/q)\t3)~(p->q)\t4)p/\(q\/r)")
        opt = int(input("Enter your choice:"))
        if(opt == 1):
            p,q  = map(int,input("Enter value for p and q:").split())
            print("~(p/\q:)",int(not(p and q)))
        elif(opt == 2):
            p,q = map(int,input("Enter value for p and q:").split())
            print("~(p\/q):",int(not(p or q)))
        elif(opt == 3):
            p,q = map(int,input("Enter value for p and q:").split())
            print("~(p->q):",int(p and (not q)))
        elif(opt == 4):
            p,q,r = map(int,input("Enter value for p ,q,r:").split())
            print("p/\(q\/r):",int(p and(q or r)))
        else:
            print("INVALID OPTION!!")
        ch = input("Continue(y/n)?")
print("\nTruth-table for De-Morgans Law")
print(ttg.Truths(['p','q'],['~(p and q)','~(p or q)' ,'~p','~q','~p or ~ q','~p and ~ q']))
print("\nTruth-table for Negation of conditional law")
print(ttg.Truths(['p','q'],['~(p=>q)','~q','p and ~q']))
print('\nTruth-table for distributive law')
```

```
print(ttg.Truths(['p','q','r'],['p and(q or r)','(p and q) or(p and r)']))
if __name__=="__main__":
main()
```

SCREENSHOTS :

```
C:\Users\Dell\OneDrive\Desktop\python>C:/Users/Dell/AppData/Local/Programs/Python/Python39/python.exe c:/Users/Dell/OneDrive/Desktop/python/mathsproject.py
1)~(p/\q)       2)~(p\/q)      3)~(p->q)       4)p/\(q\/r)
Enter your choice:1
Enter value for p and q:1 0
~(p/\q:) 1
Continue(y/n)?y
1)~(p/\q)       2)~(p\/q)      3)~(p->q)       4)p/\(q\/r)
Enter your choice:2
Enter value for p and q:1 1
~(p\/q): 0
Continue(y/n)?y
1)~(p/\q)       2)~(p\/q)      3)~(p->q)       4)p/\(q\/r)
Enter your choice:3
Enter value for p and q:0 1
~(p->q): 0
Continue(y/n)?y
1)~(p/\q)       2)~(p\/q)      3)~(p->q)       4)p/\(q\/r)
Enter your choice:4
Enter value for p ,q,r:1 1 0
p/\(q\/r): 1
Continue(y/n)?n
```

```
Truth-table for De-Morgans Law
+-----+-----+--------------+-------------+------+------+-------------+--------------+
|  p  |  q  |  ~(p and q)  |  ~(p or q)  |  ~p  |  ~q  |  ~p or ~ q  |  ~p and ~ q  |
|-----+-----+--------------+-------------+------+------+-------------+--------------|
|  1  |  1  |      0       |      0      |  0   |  0   |      0      |      0       |
|  1  |  0  |      1       |      0      |  0   |  1   |      1      |      0       |
|  0  |  1  |      1       |      0      |  1   |  0   |      1      |      0       |
|  0  |  0  |      1       |      1      |  1   |  1   |      1      |      1       |
+-----+-----+--------------+-------------+------+------+-------------+--------------+

Truth-table for Negation of conditional law
+-----+-----+-----------+------+------------+
|  p  |  q  |  ~(p=>q)  |  ~q  |  p and ~q  |
|-----+-----+-----------+------+------------|
|  1  |  1  |     0     |  0   |     0      |
|  1  |  0  |     1     |  1   |     1      |
|  0  |  1  |     0     |  0   |     0      |
|  0  |  0  |     0     |  1   |     0      |
+-----+-----+-----------+------+------------+
```

Truth-table for distributive law

p	q	r	p and(q or r)	(p and q) or(p and r)
1	1	1	1	1
1	1	0	1	1
1	0	1	1	1
1	0	0	0	0
0	1	1	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	0	0

CONCLUSION:

In this course activity we have learnt about laws of logic and how to solve basic expressions using them. Also we learnt to solve the given expression using programming language in our case python, to get the complete truth table of given expression or solve the expression for a given set of values of variable.

REFERENCES:

- https://en.wikipedia.org/wiki/Propositional_calculus#Basic_and_derived_argument_forms
- https://pypi.org/project/truth-table-generator/