

ARTIFICIAL INTELLIGENCE SESSIONAL

Course Code: CSE 4111

LAB REPORT

Department of Computer Science and Engineering

Submitted By:

Name: Safaruzzaman Shovo

Exam Roll No: 64

Reg. No: 3052

Session: 2019–20

Submitted To:

Md. Tuhin Reza

Assistant Professor, Department of CSE

Faridpur Engineering College, Faridpur

INDEX

Exp. No.	Title of Experiment	Page No.
1	Arithmetic Operations Using Python	2
2	Arithmetic Operations Using Prolog	3
3	Addition of Two Numbers Using Prolog	4
4	Sum of All Elements in a List Using Prolog	5
5	Reverse a Number Using Python	6
6	FOL Conversion and Prolog (Likes Problem)	7
7	Family Relationship Using FOL and Prolog	8

Experiment No. 1

Title: Arithmetic Operations Using Python

Objective: To perform addition, subtraction, multiplication, and division using Python.

Algorithm:

1. Input two numbers.
2. Apply arithmetic operators.
3. Display results.

Program:

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))

print("Addition=", a + b)
print("Subtraction=", a - b)
print("Multiplication=", a * b)
print("Division=", a / b)
```

Output:

```
Enter first number: 10
Enter second number: 5
Addition = 15
Subtraction = 5
Multiplication = 50
Division = 2.0
```

Conclusion: Arithmetic operations were successfully performed using Python.

Experiment No. 2

Title: Arithmetic Operations Using Prolog

Objective: To perform subtraction and multiplication using Prolog.

Program:

```
subtract(A,B,R) :- R is A-B.  
multiply(A,B,R) :- R is A*B.
```

Output:

```
?- subtract(10,5,R).  
R = 5.  
  
?- multiply(4,3,R).  
R = 12.
```

Conclusion: Prolog successfully handled arithmetic operations.

Experiment No. 3

Title: Addition of Two Numbers Using Prolog

Objective: To add two numbers using Prolog.

Program:

```
add(A,B,R) :- R is A+B.
```

Output:

```
?- add(5,7,R).  
R = 12.
```

Conclusion: Addition was implemented successfully.

Experiment No. 4

Title: Sum of Elements in a List Using Prolog

Objective: To calculate sum of list elements.

Program:

```
sumlist([],0).  
sumlist([H|T],S):- sumlist(T,S1), S is H+S1.
```

Output:

```
?- sumlist([1,2,3,4,5],S).  
S = 15.
```

Conclusion: Recursive list processing was achieved.

Experiment No. 5

Title: Reverse a Number Using Python

Objective: To reverse a given number.

Program:

```
n = int(input("Enter number: "))
rev = 0
while n > 0:
    rev = rev*10 + n%10
    n = n//10
print("Reversed number=", rev)
```

Output:

```
Enter number: 1234
Reversed number = 4321
```

Conclusion: Number reversal was successful.

Experiment No. 6

Title: FOL Conversion and Prolog Implementation (Likes Problem)

Objective: To convert given facts and rules into First Order Logic (FOL) and solve them using AI Prolog.

FOL Representation:

- $\forall x(\text{likes}(x, \text{football}) \rightarrow \text{likes}(\text{sabbir}, x))$
- $\forall y(\text{likes}(\text{sabbir}, y) \rightarrow \text{likes}(\text{sakib}, y))$

Prolog Facts:

```
likes(sakib, cricket).  
likes(sakib, football).  
likes(sakib, rugby).  
likes(riad, football).  
likes(riad, rugby).  
likes(sabbir, flower).  
likes(sabbir, custard).  
likes(sabbir, fruits).
```

Prolog Rules:

```
likes(sabbir, X) :- likes(_, football).  
likes(sakib, X) :- likes(sabbir, X).
```

Query and Output:

```
?- likes(sakib, X).  
X = flower ;  
X = custard ;  
X = fruits.
```

Conclusion: FOL statements were successfully converted and solved using Prolog.

Experiment No. 7

Title: Family Relationship Representation Using FOL and Prolog

Objective: To represent family relationships using FOL and Prolog.

Prolog Facts:

```
brother(rashid, tamim).  
brother(rashid, rishab).  
sister(champa, tamim).  
sister(champa, rishab).  
  
man(rashid).  
man(tamim).  
man(rishab).  
woman(champa).
```

Queries and Output:

```
?- woman(X).  
X = champa.  
  
?- man(X).  
X = rashid ;  
X = tamim ;  
X = rishab.  
  
?- brother(rashid, X).  
X = tamim ;  
X = rishab.
```

Conclusion: Family relationships were successfully represented and queried using Prolog.