

**St. Francis Institute of Technology  
Borivli (West), Mumbai-400103  
Department of Information Technology**

**Experiment – 9**

**1. Aim:** To install SWI Prolog

**2. Objective:** After performing the experiment, the students will be able to download and install SWI Prolog for logic programming

**3. Lab objective mapped:** To understand and implement declarative programming paradigm through logic programming (PSO2) (PO1)

**4. Prerequisite:** Nil

**5. Requirements:** The following are the requirements – Prolog Compiler

**6. Pre-Experiment Theory:**

Logic programming is a distinctive style of programming. It is based on Collection of logical propositions and questions. There are only two components to any program: facts and rules.

- The program is a knowledge base of facts and a series of rules to be applied to knowledge base
- Programs are based on the techniques developed by logicians to form valid conclusions from available evidence (knowledge base)

Most widely used logic programming language is Prolog. The name stands for **Programming in Logic**.

- Facts are statements about what is true about a problem, instead of instructions how to accomplish the solution.
- The Prolog system uses the facts to work out how to accomplish the solution by searching through the space of possible solutions.
- It is defined by an identifier followed by an n-tuple of constants.
- A relation identifier is referred to as a predicate
- When a tuple of values is in a relation we say the tuple satisfies the predicate. ▪

Rules specify under what conditions a tuple of values satisfies a predicate. ▪ The basic building block of a rule is called an *atom*

**7. Laboratory Exercise**

A. Steps to be implemented

1. Visit <https://www.swi-prolog.org/download/stable>

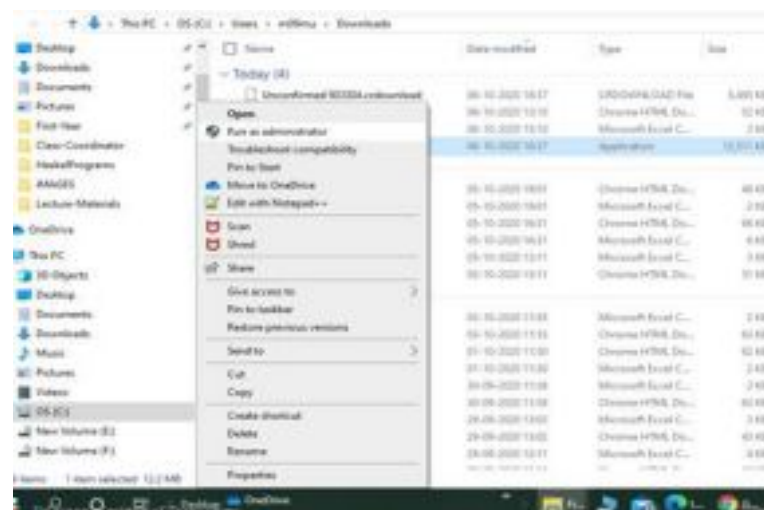
2. Click on the highlighted box



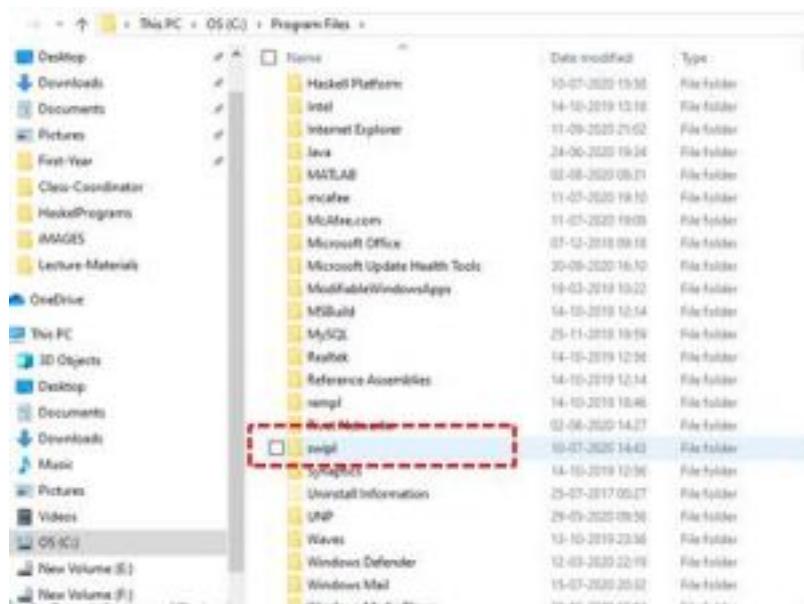
3. Download the highlighted version



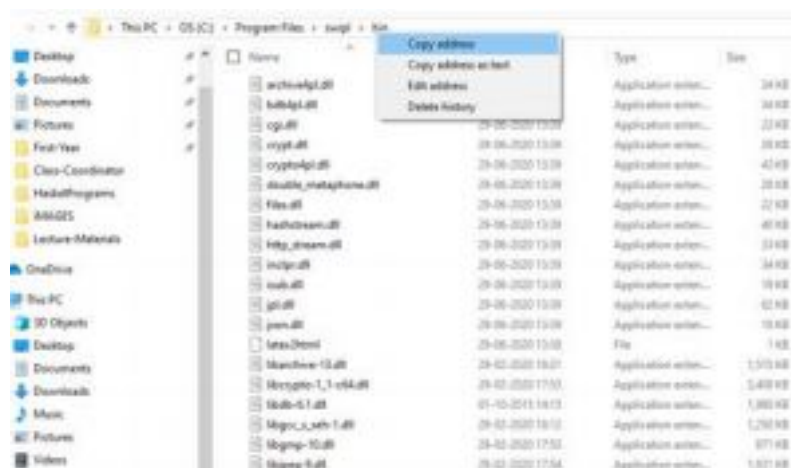
4. After successful download, install using 'Run as administrator'. This will start the installation process.



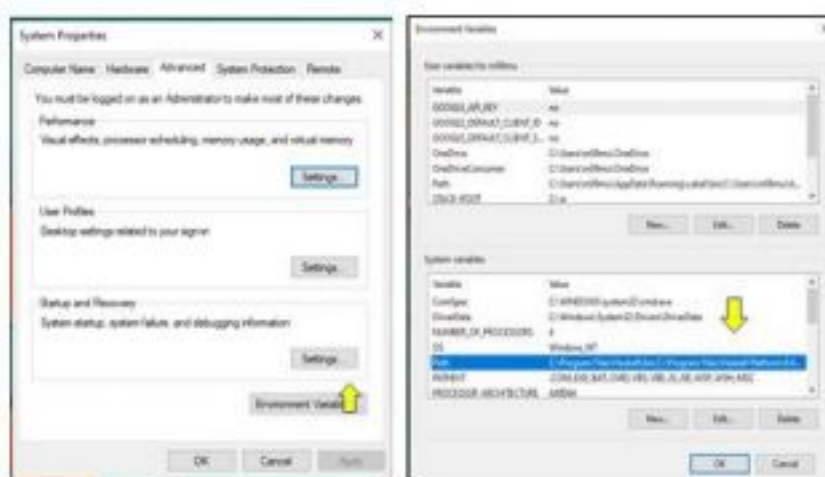
5. By default it will get downloaded in C:\program Files\Swipl folder



6. To set the path copy the entire address till bin

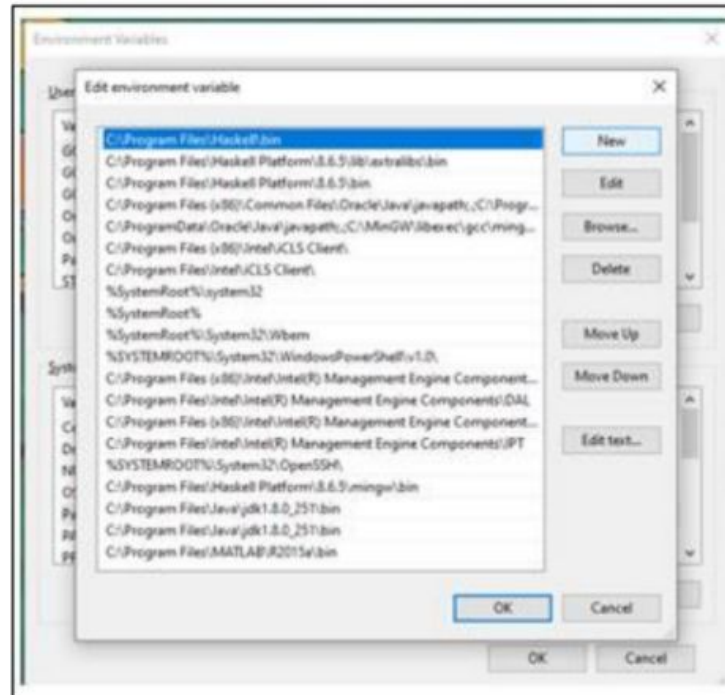


7. To set the path , go to system variables (under advanced system settings) and select the 'path' variable



Double click on path to get the new window...

8. Click on NEW and paste the entire copied path till the bin to set the path. Your path is now set successfully



9. On the desktop, double click on the SWI-Prolog Icon to open the prompt

## B. Program Code

Nil

## 8. Post Experimental Exercise

### A. Questions:

#### 1. Define facts, Rules and knowledge base

**Facts** are statements about what is true about a problem, instead of instructions how to accomplish the solution. The Prolog system uses the facts to work out how to accomplish the solution by searching through the space of possible solutions. It is defined by an identifier followed by an n-tuple of constants. A relation identifier is referred to as a predicate. When a tuple of values is in a relation we say the tuple satisfies the predicate.

Example- Valuable(gold). English meaning- Gold is Valuable, owns(john, gold). English meaning- John owns gold

**Rules** are specified in the form:

head:- t1, t2, t3, ....., tk. Where  $k \geq 1$

The head is known as the clause of head. ':-' is known as the clause neck. It is read as 'if'. The body of the clause is specified by t1, t2, t3, tk. It contains one or more components, and it can be separated using the commas. The goal represents the components. The command is represented by 'and'. A rule will be read as 'if t1, t2, t3, ....., tk are all true, head is true'.

Example:

large\_animal(X):-animal(X),large(X).

grandparent(X,Y):-father(X,Z),parent(Z,Y).

go:-write('hello world'),nl.

Knowledge base is the collection of all the facts and rules entered by the user.

## 2. Highlight important features of logic programming

Logic programming is a Distinctive style of Programming. It is a Collection of logical propositions and questions. There are only two components to any program: facts and rules. The program is a knowledge base of facts and a series of rules to be applied to the knowledge base. Programs are based on the techniques developed by logicians to form valid conclusions from available evidence (knowledge base). Interaction with the program is done via posing queries to an inference engine (also called a query interpreter)

## 3. Differentiate between logic and imperative programming paradigm

| Logical Paradigm  | Imperative Paradigm  |
|---|--|
| Logical programming is a programming paradigm that expresses the logic of a computation without describing its control flow.  | Imperative programming is a programming paradigm that uses statements that changes the program's state.  |
| Program statements are expressed as facts and rules about problems within a system of formal logic.   | Imperative programming paradigm consists of several statements, and after the execution of all of them, the result is stored. It's about writing a list of instructions to tell the computer what to do step by step.  |
| Logical programming focuses on what the program should accomplish.  | Imperative programming focuses on how the program should achieve the result.   |
| Example: To print sum of 1 to 10<br><code>sumTo(1,1).</code><br><code>sumTo(N,M):-N&gt;1,</code><br><code>N1 is N-1,</code><br><code>sumTo(N1,M1),</code><br><code>M is M1+N.</code><br><code>?- sumto(10,N).</code><br><code>N = 55</code> | Example: To print sum of 1 to 10<br><code>#include &lt;stdio.h&gt;</code><br><code>int main(){</code><br><code>int sum = 0;</code><br><code>sum += 1;</code><br><code>sum += 2;</code><br><code>sum += 3;</code><br><code>sum += 4;</code><br><code>sum += 5;</code><br><code>sum += 6;</code><br><code>sum += 7;</code><br><code>sum += 8;</code><br><code>sum += 9;</code><br><code>sum += 10;</code><br><code>printf("The sum is: %d\n", sum);</code><br><code>return 0;</code><br><code>}</code> |

**B. Results/Observations/Program output:** Nil

### **C. Conclusion:**

In this experiment we have successfully downloaded and launched SWI-Prolog. We have also studied various features of the prolog language and also have compared it with the imperative programming paradigm to understand the differences between the two.

To perform this experiment we have used a SWI-Prolog 8.2.3-1.

Logic programming is a type of computer programming that is based on formal logic rules. In a sense, logic programming corresponds to the use of mathematical equations and other logical constructs to drive programming outcomes. Rules are written as logical clauses with a head and a body; for instance, "H is true if B1, B2, and B3 are true." Facts are expressed similar to rules, but without a body; for instance, "H is true."

### **D. References:**

- [1] Michael L Scott, "Programming Language Pragmatics", Third edition, Elsevier publication
- [2] Max Bramer, " Logic Programming with Prolog", Springer, 2005