

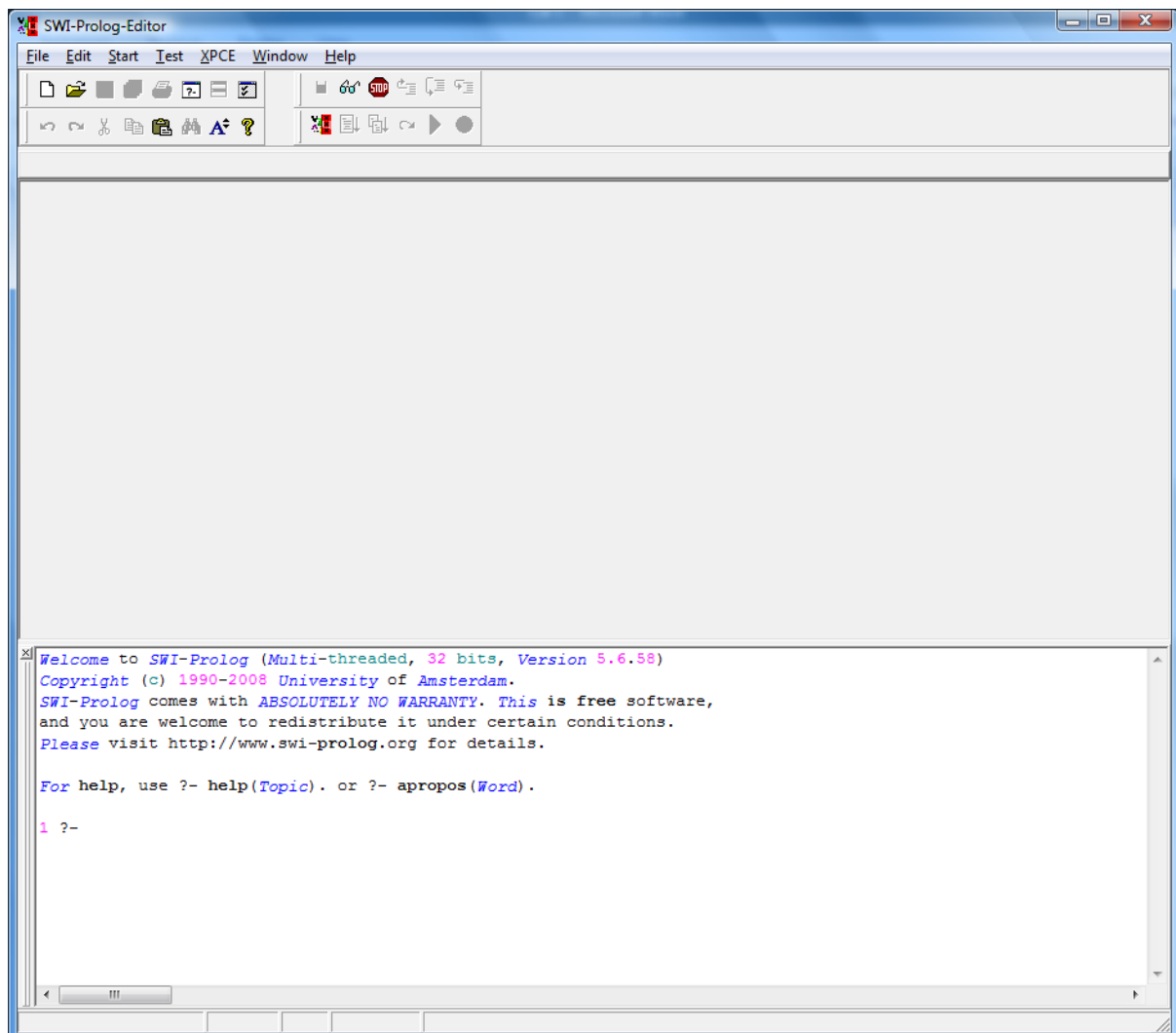
Lab 1 – 08226 Artificial Intelligence

This lab tutorial will introduce you to the Prolog Editor IDE which will be used throughout this module, and to using facts in Prolog.

1.0 SWI-Prolog-Editor


The Prolog compiler has already been installed on the lab PCs, but we need to install the Prolog Editor for us to use. Download **prolog_editor.zip** from the Software section of the module SharePoint site. Extract the contents to the folder **G:/08226/Prolog_Editor/**.

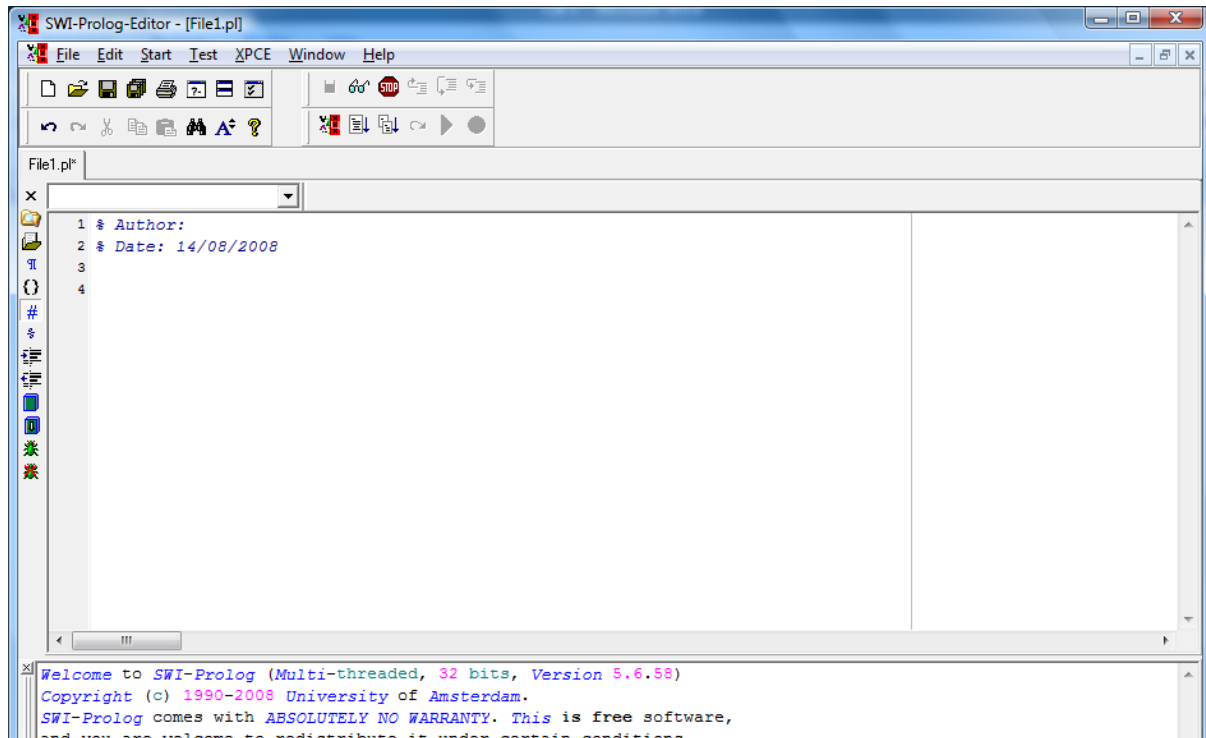
Start SWI-Prolog-Editor (**G:/08226/Prolog_Editor/SwiplEdit.exe**) and you will be presented with the following program:




This is the Prolog IDE that you will use to write your Prolog programs. The IDE is split into three sections, namely the Menu and Tool Bar section, the Code Preview section, and the Query Window section.


2.0 Creating a New Prolog Program

Select the New Button  from the Tool Bar and you will see a blank page in the Code Preview section.



SWI-Prolog-Editor has created a new Prolog program for us. The first two lines of code are comments that you can use or delete. There are two ways of adding comments into your programs. You can use the % symbol as above or you can encapsulate areas of code as comments between /* and */ just as you would in C#.

The first thing we need to do is save our program to our file store. Select the Save Button  and then save the prolog file as **Lab1.pl** to the location **G:/08226/Lab 1/**.

Our program does not do anything but let us try to compile our program. Select the Consult Button (Compile Button) . You should see a message stating something like:

```
% G:/08226/Lab 1/Lab1.pl compiled 0.00 sec
true.
```

*[If you do not see the above message then click inside of the Query Window and press **Esc**, and then try again.]*

3.0 Prolog Basics

A prolog program is essentially a database of facts and rules. Facts are simply logical facts and rules are similar to functions or methods.

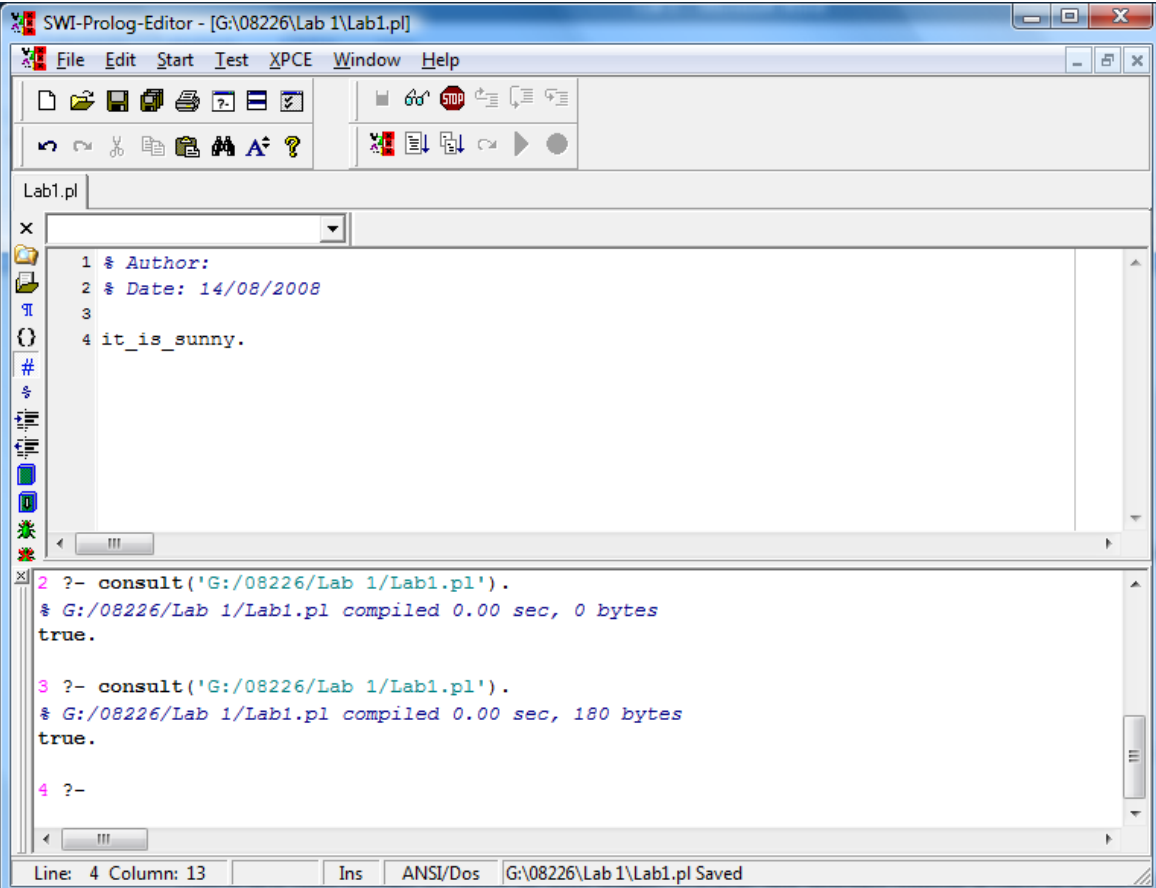
3.1 Simple Facts

We are going to add the fact that **it is sunny** to our program. We do this as follows:

```
it_is_sunny.
```

There are two things to note at this stage. Firstly, we always follow a fact with a full stop (period). Secondly a fact name can only start with a lower-case letter, so we are not allowed to add a fact name that starts with an upper-case or a number.

Add the fact to your program, save your program and then select **consult** (compile). You should see the following:



The screenshot shows the SWI-Prolog-Editor window with the title bar "SWI-Prolog-Editor - [G:\08226\Lab 1\Lab1.pl]". The menu bar includes File, Edit, Start, Test, XPCE, Window, and Help. The toolbar contains various icons for file operations, editing, and execution. The main text area shows the source code for Lab1.pl:

```
1 % Author:
2 % Date: 14/08/2008
3
4 it_is_sunny.
```

The console window at the bottom shows the following output:

```
2 ?- consult('G:/08226/Lab 1/Lab1.pl').
   % G:/08226/Lab 1/Lab1.pl compiled 0.00 sec, 0 bytes
true.

3 ?- consult('G:/08226/Lab 1/Lab1.pl').
   % G:/08226/Lab 1/Lab1.pl compiled 0.00 sec, 180 bytes
true.

4 ?-
```

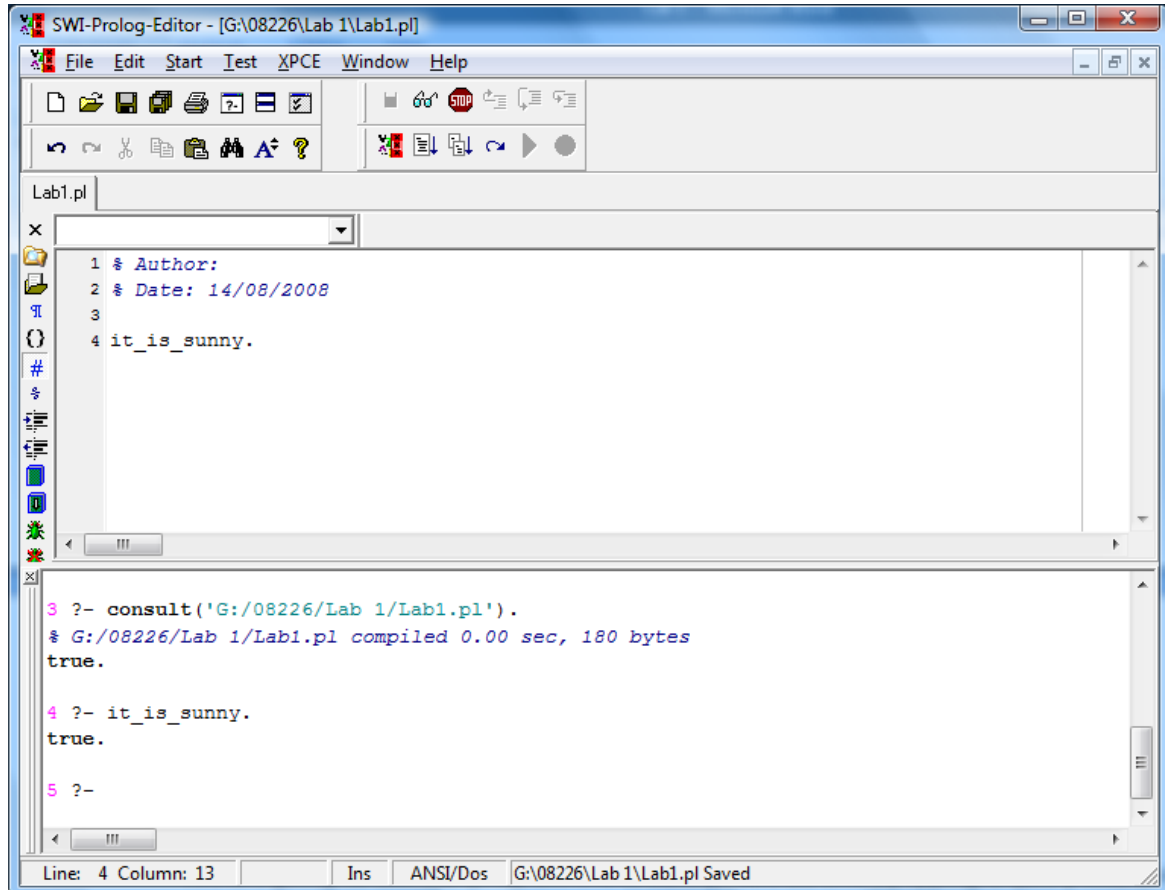
The status bar at the bottom indicates "Line: 4 Column: 13", "Ins", "ANSI/Dos", and "G:\08226\Lab 1\Lab1.pl Saved".

3.2 Queries

We are going to ask our program if **it is sunny**. We do this by typing a query into the Query Window. In the Query Window type the following followed by the enter key:

```
?- it_is_sunny.
```

You should see the following:



Prolog has answered **true**. This is because the fact (**it_is_sunny**) was found in our program database.

We are going to ask our program if **it is raining**. We do this by typing a query into the Query Window. In the Query Window type the following followed by the enter key:

```
?- it_is_raining.
```

You should see that Prolog has either generated an **error** or has answered **false** (depending on the version of Prolog you are using). This is because we do not have the fact **it_is_raining** in our program database.

3.2.1 Queries Test Exercise

Write down your answers to the following questions without using Prolog, and then check your answers at the end of this document and/or using Prolog.

Which of the following are correctly defined facts?

- | | |
|------------------------|----------------------------|
| 1) <code>sunny.</code> | <i>true / false</i> |
| 2) <code>fOgGy.</code> | <i>true / false</i> |
| 3) <code>Foggy.</code> | <i>true / false</i> |
| 4) <code>2day.</code> | <i>true / false</i> |
| 5) <code>day_2.</code> | <i>true / false</i> |

Given the following Prolog program, which queries will return true?

```
red_box.  
blue_square.  
yellow_triangle.  
blue_cube.  
red_line.
```

- | | |
|-------------------------------------|------------------------|
| 6) <code>?- yellow_triangle.</code> | <i>yes / no</i> |
| 7) <code>?- red_square.</code> | <i>yes / no</i> |
| 8) <code>?- red_box.</code> | <i>yes / no</i> |
| 9) <code>?- blue_square.</code> | <i>yes / no</i> |
| 10) <code>?- blue_line.</code> | <i>yes / no</i> |

Check your answers here: [Queries Test Exercise Answers](#)

3.3 Facts with Arguments

We are going to add some facts that have some form of relationship. We do this in the form:

```
relation(<argument1>,<argument2>,...,<argumentN>).
```

Like simply facts, relation names must begin with a lower-case letter.

Facts are described by their number of arguments, e.g.

<code>relation.</code>	would be described as relation/0
------------------------	---

<code>relation(argument1).</code>	would be described as relation/1
-----------------------------------	---

<code>relation(argument1,argument2).</code>	would be described as relation/2
---	---

The following are some facts that describe relationships:

```
car(ferrari).
```

This fact is stating the relationship that a **ferrari** is a **car**.

```
red(ferrari).
```

This fact is stating the relationship that **ferrari** is **red**.

```
likes(darren,beer).
```

This fact is stating the relationship that **darren** likes **beer**.

Add the above three facts into our program, and then select consult (compile).

Ask Prolog the following query:

```
?- car(ferrari).
```

You should have the answer **true** since we have this fact in our program database.

Ask Prolog the following query:

```
?- car(mazda).
```

You should have the answer **false** since we do not have this fact in our program database.

3.3.1 Facts with Arguments Test Exercise

Write down your answers to the following questions without using Prolog, and then check your answers at the end of this document and/or using Prolog.

Given the following Prolog program, which queries will return true?

```
age(darren,21).  
age(paul,34).  
age(dawn,30).  
age(steve,14).  
age(katey,2).
```

- | | |
|-----------------------|---------------------|
| 1) ?- age(dawn,30). | true / false |
| 2) ?- age(paul). | true / false |
| 3) ?- age(14,steve). | true / false |
| 4) ?- age(34). | true / false |
| 5) ?- age(katey,two). | true / false |

Check your answers here: [Facts with Arguments Test Exercise Answers](#)

3.4 Queries Test Exercise Answers

Which of the following are correctly defined facts?

- | | |
|------------------------|---|
| 1) <code>sunny.</code> | true – the fact name starts with a lower-case letter |
| 2) <code>fOgGy.</code> | true – the fact name starts with a lower-case letter |
| 3) <code>Foggy.</code> | false – the fact name starts with an upper-case letter |
| 4) <code>2day.</code> | false – the fact name starts with a number |
| 5) <code>day_2.</code> | true – the fact name starts with a lower-case letter |

Given the following Prolog program, which queries will return true?

```
red_box.  
blue_square.  
yellow_triangle.  
blue_cube.  
red_line.
```

- | | |
|----------------------------------|--|
| 6) <code>yellow_triangle.</code> | yes – the fact is defined in the program |
| 7) <code>red_square.</code> | no – the fact is not defined in the program |
| 8) <code>red_box.</code> | yes – the fact is defined in the program |
| 9) <code>blue_square.</code> | yes – the fact is defined in the program |
| 10) <code>blue_line.</code> | no – the fact is not defined in the program |

Check the answers using Prolog.

Return to your exercise here: [Queries Test Exercise](#)

3.5 Facts with Arguments Test Exercise Answers

Given the following Prolog program, which queries will return true?

```
age(darren, 21) .  
age(paul, 34) .  
age(dawn, 30) .  
age(steve, 14) .  
age(katey, 2) .
```

- | | |
|-------------------------|--|
| 1) ?- age(dawn, 30) . | true – this fact is defined in the program |
| 2) ?- age(paul) . | false – this fact is not defined in the program |
| 3) ?- age(14, steve) . | false – this fact is not defined in the program |
| 4) ?- age(34) . | false – this fact is not defined in the program |
| 5) ?- age(katey, two) . | false – this fact is not defined in the program |

Check the answers using Prolog.

Return to your exercise here: [Facts with Arguments Test Exercise](#)