
Programming Paradigms – Setup

FS 2019

Tutorial: Setting up the Environment

Session: 08.03.2019

Note: Attendance at the exercise session on the 8th of March is not mandatory and you will not have to hand in or show any part of this tutorial, however, you will need a working C++ compiler and working interpreters for Haskell, Prolog and Python to be able to successfully complete the exercises. If you have any questions or need help with installation you are recommended to take part in the tutorial session or otherwise send an email to maarten.schenk@unibas.ch or m.pluess@unibas.ch.

Question 1: Introduction

(0 points)

This is a quick guide for installing the tools required in this course. You are free to use the environment you like most. This introduction further explains what is needed to install the tools on your system.

a) Linux

For Linux the installed package manager should provide everything needed.

b) macOS

For macOS we suggest that you install *homebrew*. In a nutshell, homebrew is an easy to use package manager similar to package managers known on Linux platforms. To install it, execute the following command in the Terminal:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

You can also visit the official website <https://brew.sh> to copy this command or to learn more if you want to install homebrew manually. As an alternative to *homebrew*, you can also use *MacPorts*.

c) Windows

For Windows we suggest that you use the *Cygwin* environment. Cygwin provides a UNIX-like environment for Windows. Download and launch the installer <https://www.cygwin.com>. During the installation process you will be presented a list with all available packages to install. After the installation you can rerun the installer to add additional packages.

If you have a 64bit installation of the *Windows 10 Anniversary Update* or newer and you would like to follow the Linux instructions you can install the Linux subsystem for Windows. To find out how to install the Linux subsystem for Windows you can go to https://msdn.microsoft.com/en-us/commandline/wsl/install_guide for an official guide from Microsoft.

d) IDEs

We recommend using the compilers and interpreters mentioned in this document without the use of a real IDE. However, you may write your code in whichever editor you like, as long as the final code compiles/runs with the compilers/interpreters we suggest.

Question 2: C++

(0 points)

This section explains how you can install the GNU C++ *g++* compiler and the tools for Eclipse. To test your installation, you can use the following hello-world program:

```
// hello.cpp
#include <iostream>
int main(int argc, const char** argv)
{
    std::cout << "Hello C++ World!" << std::endl;
}
```

If you are compiling the program through the command line, compile it using `g++ hello.cpp -o hello` (where `hello.cpp` is the source file and `-o hello` specifies the executable to be generated).

Then, run it using `./hello`.

a) Linux

Run the following command in a terminal:

```
sudo apt-get install build-essentials
```

b) macOS

If you haven't installed XCode yet, install it (it is technically possible to install the command line tools without XCode, but an XCode installation is advised). After installation is complete, open XCode at least once to complete first time setup. Once initial setup has finished, open a terminal window and run:

```
xcode-select --install
```

Follow the instructions in the window that pops up to install the command line tools, which include the `g++` compiler.

c) Windows

Find the `gcc-g++` package in the cygwin installation and install it.

Question 3: Haskell

(0 points)

For the Haskell exercises, we will use the *Glasgow Haskell Compiler (ghc)*. For this course, it is sufficient to only install the compiler `ghc` itself.

If you like, you can also install the entire *Haskell Platform*. It contains additional tools, such as *Cabal* and *Scion-Browser*.

To test your installation, you can use the following hello world program:

```
-- hellohaskell.hs
module Main where

main::IO()
main = putStrLn "Hello, Haskell World!"
```

Start it directly using the Haskell interpreter: `ghci hellohaskell.hs`. This gives you a haskell command prompt, where you type in `main` to launch the helloworld program. The command prompt can be closed using CTRL-Z.

Alternatively, you can also compile it into a regular executable using `ghc hellohaskell.hs -o hellohaskell`. In that case, launch it directly using `./hellohaskell`.

a) Linux

```
sudo apt-get install ghc
or
sudo apt-get install haskell-platform
```

b) macOS

```
brew install ghc
or
brew install haskell-platform
```

c) Windows

The *Haskell Platform* can be downloaded from the Haskell homepage (<https://www.haskell.org>). The binaries should be automatically linked to your Cygwin environment.

Question 4: Prolog

(0 points)

For Prolog we use the SWI-Prolog (<http://www.swi-prolog.org>) implementation. To test your installation, use the following hello-world program (note that the file name must start with a lowercase letter):

```
% helloprolog.pl
hello :- write('Hello Prolog World!').
```

Depending on your system, you can now use the `swipl` or the `pl` command to open a Prolog console. Inside that console, run the hello world program using the steps below. Note that the periods (`.`) are also part of the commands.

1. Type `consult(helloprolog).` to load the `helloprolog.pl` application.
2. Type `hello.` to call the hello procedure.
3. Finally, close the command line using `halt.` or by pressing CTRL-Z.

a) Linux

```
sudo apt-get install swi-prolog
```

b) macOS

```
brew install swi-prolog
```

c) Windows

Find the `swi-prolog` package in the Cygwin installation.

Question 5: Python

(0 points)

For Python we use Python 3 (<https://www.python.org>). While Python 2.7 is still often used, we use Python 3 in the exercises because it is the much more modern version. To test your python installation, start the interpreter from the terminal using the command `python3` and enter:

```
print('Hello World!')
```

To exit the Python interpreter enter `quit()` or press `ctrl + D`.

a) Linux

Python 3 should be preinstalled.

b) macOS

```
brew install python3
```

c) Windows

Find the `python` package in the Cygwin installation, or download the installer from <https://www.python.org>.

Congratulations, you are now ready to start with the exercises!