



Introduction to Programming

Tutorial Task 1.1: Hello World

Overview

As a first step, create the classic “Hello World” program. This will help ensure that you have all of the software installed correctly, and are ready to move on with creating other programs.

Purpose: Install and test the tools needed to get started programming.

Task: Create your own Hello World program using the command line compiler.

Time: This task should be completed in your first lab class and submitted for feedback before the start of week 2.

- Resources:**
- Chapter 1 of the Programming Arcana
 - Swinburne CodeCasts ([YouTube Channel](#), [iTunesU](#))
 - [Series Introduction](#)
 - Install videos for [Linux](#), [Mac OS X](#), and [Windows](#).
 - [Compiling and Using the Terminal](#)
 - Syntax Videos
 - [Introduction](#), [Getting Started](#), [Calling Procedures](#), and [Creating Your Own Procedures](#)

Submission Details

You must submit the following files to Doubtfire:

- Hello World source code (HelloWorld.pas)
- Screenshot of the Terminal showing use of `cd`, `ls`, and `fpc` commands as well as execution of your Hello World program.

Make sure that your task has the following in your submission:

- Code layout - match the example for indentation and use of case (UPPERCASE, lowercase, and PascalCase) for the different aspects of the code.
- The code must compile and the screenshot show it working on your computer.

Instructions

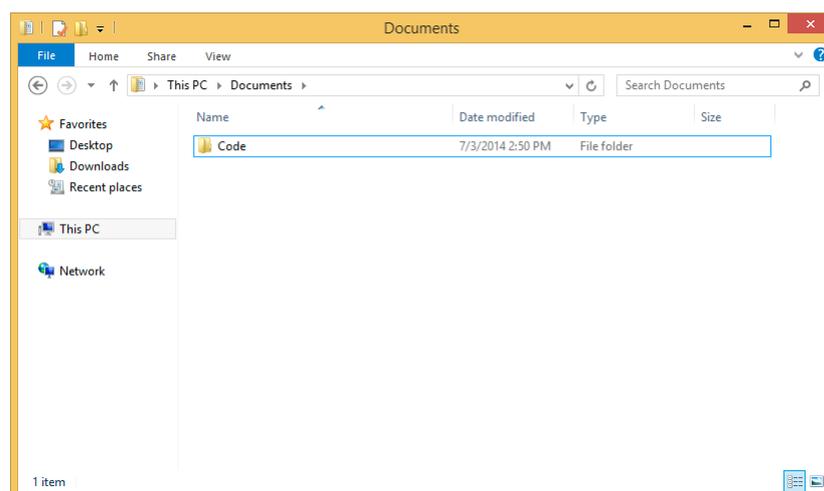
The first task includes the steps needed for you to install the tools you will need in this unit. You will then use these tools to create the classic '*Hello World*' program.

1. Install the tools you need to get started. Watch the installation video for your Operating System and *Read Chapter 1 of the Programming Arcana* for full instructions. Ensure that you have:
 - Installed MinGW for Windows, Xcode for Mac, and devtools for Linux
 - Installed the **Free Pascal Compiler** (the compiler)
 - Installed **SublimeText** (the text editor)

Note: You can skip this step on the computers in the Swinburne lab as this will already be setup.

Hint: If you are using your own Windows machine remember to install MinGW to give you access to the Terminal. See Chapter 1 of the Programming Arcana and the install video for extra help on how to do this.

2. If you don't already have one, make a directory (i.e., a 'folder') to store your code (e.g., *Documents/Code/Lab1*). On a Swinburne computer you may wish to use a directory on your student drive or a USB storage device.
 - Navigate to your *Documents* directory in Finder or File Explorer
 - Right click in the *Documents* directory and select **New Folder**, name it **Code**



3. Open **Sublime Text**, and create a new file.
4. Enter the text for the *Hello World* program. It should appear shown here.

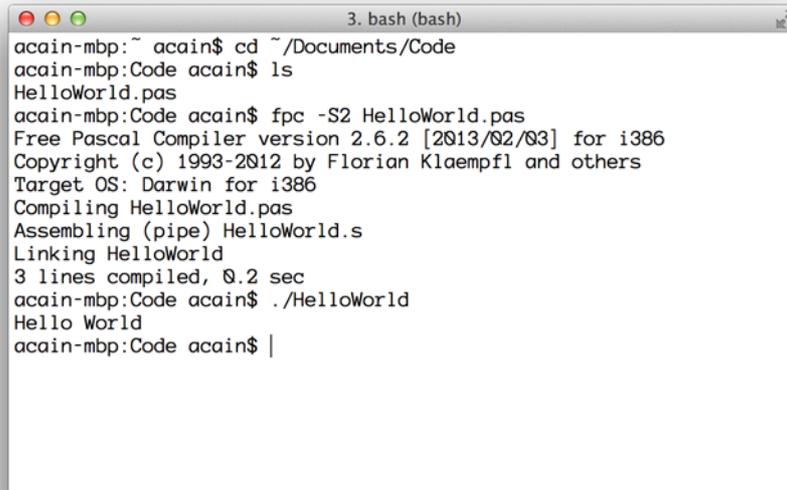
A screenshot of the Sublime Text editor window. The window title is 'HelloWorld.pas'. The editor shows four lines of Pascal code: '1 program HelloWorld;', '2 begin', '3 WriteLn('Hello World');', and '4 end.'. The code is color-coded: 'program' is green, 'begin' is red, 'WriteLn' is red, 'Hello World' is yellow, and 'end' is red. The status bar at the bottom indicates 'Line 1, Column 1', 'Tab Size: 4', and 'Pascal'.

Note: Sublime Text has a number of different colour schemes. Your colours may be different, but Sublime Text should highlight different parts of your code in different colours once you save it as a **.pas** file.

5. Save the file as **HelloWorld.pas** in your code directory.

Note: The pas file contains the *source code* for your program. To make this a *real* program you need to compile it.

6. Open a **Terminal** (MSYS or MinGW shell in Windows), then perform the following commands:
 - Change into the directory containing your code using the **cd** command.
`cd /c/Users/your_user/Documents/Code` on Windows¹ or
`cd ~/Documents/Code` on MacOS or Linux
 - List the files in this directory using the **ls** command
 - Print the working directory using the **pwd** command
 - Compile your program using **fpc -S2 HelloWorld.pas**
 - List the files in this directory using the **ls** command to see the files created by the compile process
 - Run your program using **./HelloWorld**



```
3. bash (bash)
acain-mbp:~ acain$ cd ~/Documents/Code
acain-mbp:Code acain$ ls
HelloWorld.pas
acain-mbp:Code acain$ fpc -S2 HelloWorld.pas
Free Pascal Compiler version 2.6.2 [2013/02/03] for i386
Copyright (c) 1993-2012 by Florian Klaempfl and others
Target OS: Darwin for i386
Compiling HelloWorld.pas
Assembling (pipe) HelloWorld.s
Linking HelloWorld
3 lines compiled, 0.2 sec
acain-mbp:Code acain$ ./HelloWorld
Hello World
acain-mbp:Code acain$ |
```

Tip: Bash commands (e.g., `cd`, `ls`, `pwd`, `fpc`) do not like spaces in directory or file names (e.g., `My Documents`, or `Hello World.pas`). If you have a space in the name of something you need to add in a reverse slash:

`My\ Documents` and `Hello\ World.pas`

Avoid spaces in the names of your files and folders!

¹ Replace your_user with your computer user name

Now that the Task is complete you can submit it for assessment, which will help prepare it for your portfolio.

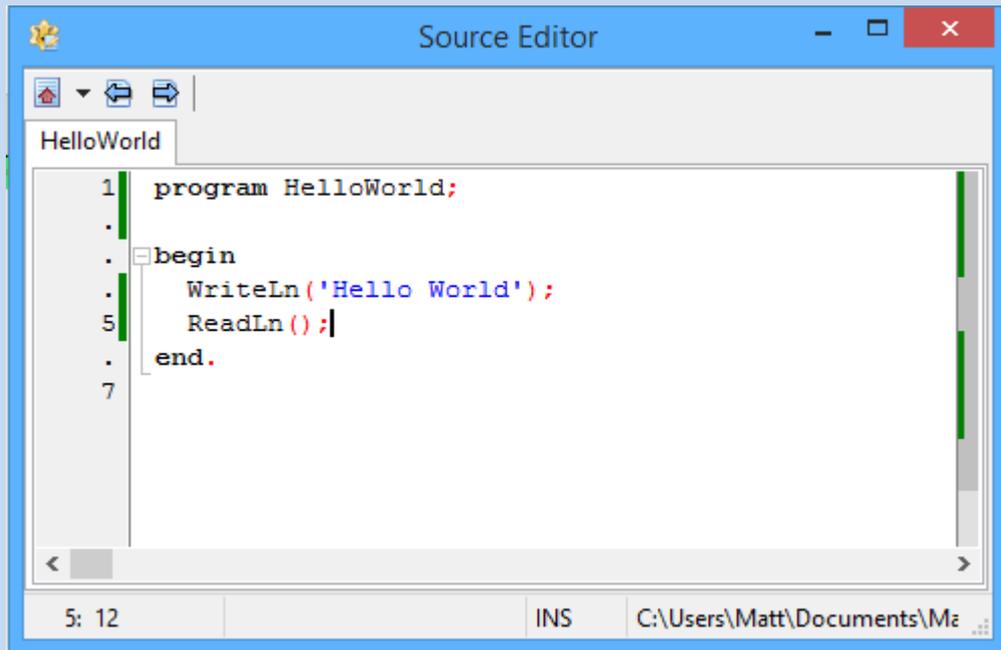
7. Use [Skitch](#) (or your preferred screenshot program) to take a screenshot of the Terminal, as this is one of the things you will need to submit.
 8. Login to Doubtfire, and locate Tutorial Task 1.1
 9. Change the status of the task to **Ready To Mark**
 10. Upload your completed Hello World code and the screenshot.
 11. If you check back later Doubtfire will have prepared these as PDFs for your tutor to assess.
-
12. Now, remember to save the document and **backup** your work to multiple locations!
 - Once you get things working you **do not** want to lose them.
 - Work on your computer's storage device most of the time... but backup your work when you finish each task.
 - Use **Dropbox** or a similar online storage provider, as well as other locations.
 - Doubtfire is not a Backup of your work, so make sure you keep a copy!
 - A USB keys and portable hard drives are good secondary backups... but can be lost/damaged (do not rely upon them).

You now have your first portfolio piece. This will help demonstrate your learning from the unit.

Note: This is one of the tasks you need to **submit to Doubtfire**. Check the assessment criteria for the important aspect your tutor will check.

Optional Extension Task:

1. Open **Lazarus** and select "Project", "New project" then "Simple Program".
2. Enter the text in the source editor for the HelloWorld program. It should look as follows:

A screenshot of the Lazarus Source Editor window. The window title is "Source Editor". The editor shows a Pascal program named "HelloWorld" with the following code:

```
1 program HelloWorld;  
.  
.  
begin  
.  
  WriteLn('Hello World');  
5  ReadLn();  
.  
end.  
7
```

The status bar at the bottom shows "5: 12", "INS", and "C:\Users\Matt\Documents\Ma".

In Lazarus select "Run" then "Run" to compile and run your program.