Week 01, Lab 01	Weight: 1%	Due: End of your stream's week 1 lab session (via `sync`)

Pre-lab Preparation:

- **Lecture: Wk01 Day 001 Introduction**

Lab Activities:

Exercise 1: Obtain, setup and power on your Raspberry Pi

The lab TAs and Technical staff will issue you with a Raspberry PI Kit.

Follow the instructions included in the Raspberry Pi Kit to plug in and power up your Raspberry Pi.

Log in as `pi`.

Connect to the AUT WiFi network using `mesa_connect`.

Register your Pi using `register`.

Login for the first time using your new username and password.

You should now see the command prompt, something like:

```
[ok] login@raspberrypi: ~$
```

This is your chance to start running some commands. The first and most import to try is `sync`.

That should download your initial lab and homework materials.

Let's try another command. The command-line (which is really a program called Bash) has quite a powerful programming language of its own, often called a *shell scripting language*. If you enter a command at the prompt it will be executed straight away. Try:

```
echo James is great
```

Read the document Bash/Mesa Command Line Guide that you have been given.

As your first challenge, navigate to the directory `~/p1.2015s1/lab01/`

List the files there using `ls` and `ls -l`

Run the command: `fbgs lab01.pdf` to view the lab tasks (this document).

Exercise 2: Drawing with the Turtle

Using the `cd` command change to the directory:

`~/p1.2015s1/lab01/working_copy/ex02/`

Use the `ls` command to list the directories contents. Check that the `lab01ex02.c` file is present. Open `lab01ex02.c` in the text editor using the following command: `editor lab01ex02.c`

First try moving the cursor around the screen.

Text that you type appears at the cursor. Try typing some text, then delete it again.

Press the `F1` key to see the editor help

Type the following program source code into the editor, and save the file (`CTRL-S`):

```
#include "p1student.h"

int main(int argc, char* argv[])
{
    create_turtle_world();

    pen_colour(MAGENTA);
    forward(200);
    turn(120);

    pen_colour(CYAN);
    forward(200);
    turn(120);

    pen_colour(YELLOW);
    forward(200);
    turn(120);

    return (p1world_shutdown());
}
```

The shortcut `CTRL-Q` can be used to quit the text editor after saving.

Check the contents of the file using the `less` command. Review the source code, ensure you have typed it in correctly. What do you expect the program to do when you execute it?

Compile the source code using the command: `make`

Run the `ls` command again, check the directory contents. How has it changed?

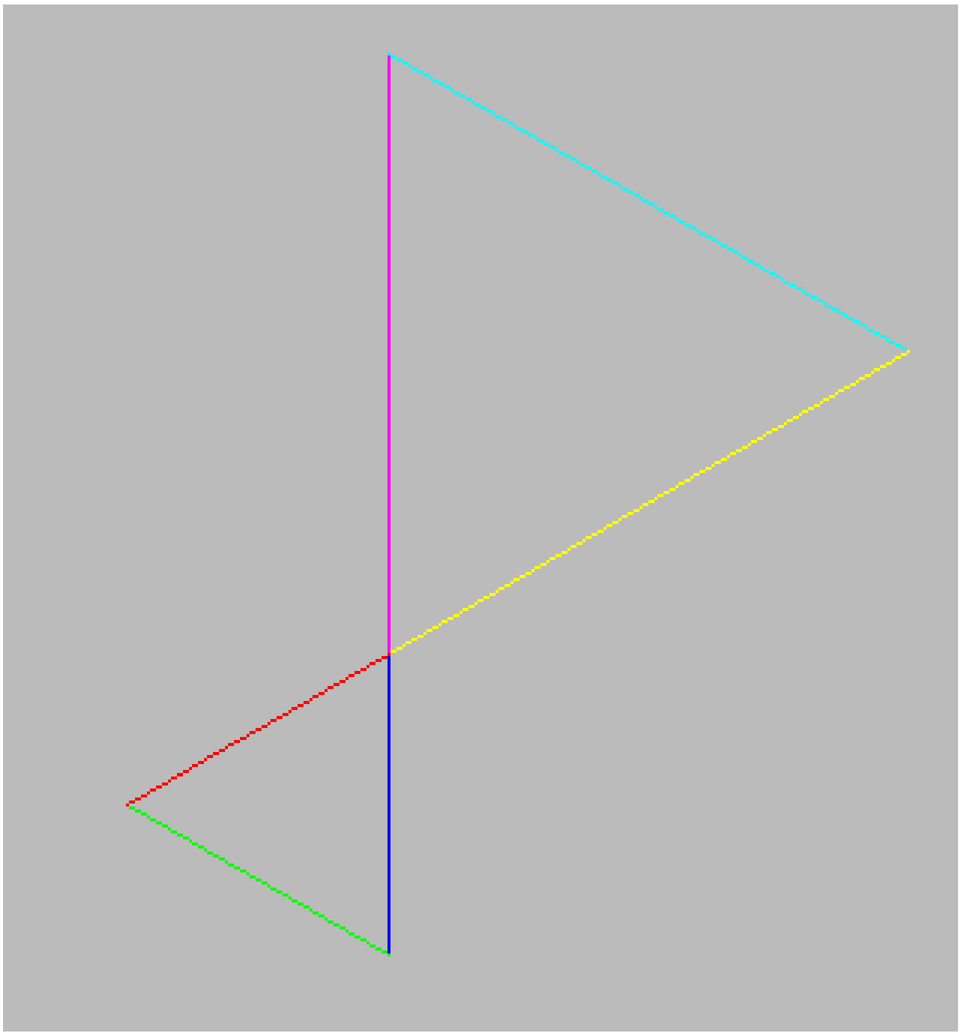
Run the program using the command: `./lab01ex02`

Exercise 3: Draw Two Triangles

The `cp` command is used for copying files. From the exercise 2 directory, copy the `lab01ex02.c` source code file into the exercise 3 directory. Name the copied file `lab01ex03.c`. You can do this by using the `cp` command as follows: `cp lab01ex02.c ../ex03/`

Now navigate to the directory: `~/p1.2015s1/lab01/working_copy/ex03/`

Open `lab01ex03.c` in the text editor. The file should contain the code to draw the single triangle seen in the previous exercises. Modify the source code to create the following output, you will need to add more turtle commands in the correct sequence to achieve the desired result:



Compile and run your program to test the source code. Does your output match the required output? If it makes, move onto the next exercise. Otherwise, re-open the source code with the text editor and edit your code until your program achieves the required output. Think carefully about the sequence of commands required to achieve the desired output drawing of the two triangles.

Exercise 4: Coding Standards, Whitespace

Navigate to the directory: ~**`/p1.2015s1/lab01/working_copy/ex04/`**

Compile and run this program using **`make`**. Note what the turtle outputs.

Open **`lab01ex04.c`** in the text editor. Notice the code in this source file is extremely messy! Use good white spacing practices to tidy the source code. Code should be indented with a tab, or four spaces as seen in the code sample in exercise 2. Ensure you keep the braces **`{`** and **`}`** in the same style as seen in exercise 2. Overall, ensure the code is human readable.

Check the program still compiles and runs after editing. Is it still outputting the same turtle drawing after tidying the code as it was before you started editing?

Exercise 5: Experiment with the Turtle World

Navigate to the directory: ~**`/p1.2015s1/lab01/working_copy/ex05/`**

Open **`lab01ex05.c`** in the text editor. Write some turtle commands to draw something creative... impress us with the turtle graphics skills you have learnt so far!

Week 01, Lab 01 Submission:

Run the **`sync`** command to submit your completed lab work.

Shutdown your Raspberry PI by pressing **`ALT-CTRL-DEL`**. Power-down and pack up your Raspberry Pi kit.

Marking Criteria:

Have you completed each of the following? Have you submitted your code from lab?

Marking Criteria:	**Week 01 Lab 01** **Weight 1%**	**Yes**	**No**
Ex 1:	Raspberry Pi obtained and setup correctly?		
	Raspberry Pi mesa user account created?		
Ex 2:	Source code compiles, runs and draws correctly?		
Ex 3:	Program draws two triangles are required?		
Ex 4:	Whitespace has been tidied?		
	Program still compiles and runs correctly?		
Ex 5:	Personal creative free drawing program completed?		
	Total:		

Next activity: Final Week 1 Lecture and Week 1 Homework.