Week 04, Lab 04	Weight: 1%	Due: End of your stream's week 4 lab session (via `sync`)

Pre-lab Preparation:

- **Week 1, 2, 3 Lectures, Week 1, 2, 3 Labs**
- **Lecture: Wk04 Day 010, 011**

Lab Activities:

Remember to **`sync`** to obtain the lab starting code.

Exercise 1: Basic printf usage

Navigate to the directory: **`~/p1.2015s1/lab04/working_copy/ex01/`**

Write a program using **`printf`** that outputs the following at the console, ensure your programs output accurately replicates the following:

```
+-----------------------------------------------+
| Console Programming on the Raspberry PI with C |
+-----------------------------------------------+
```

Open **`lab04ex01.c`** in the editor and begin! Be sure to test your program as you develop. Edit, compile and test often as you add features. Remember use make to compile your program.

Exercise 2: sizeof the C types

Navigate to the directory: **`~/p1.2015s1/lab04/working_copy/ex02/`**

Write a program using **`printf`** and **`sizeof`** that outputs the size in bytes of each primate type available in C at the console. Open **`lab04ex02.c`** in the editor and begin!

Each type should output a line of text such as:

```
The size of a char is 1 byte(s).
```

Once your program runs correctly, complete the table below based upon the output of your program:

Data Type	Size in bytes on Pi
`char`	
`unsigned char`	
`signed char`	
`short`	
`unsigned short`	
`int`	

Data Type	Size in bytes on Pi
`unsigned int`	
`long`	
`unsigned long`	
`float`	
`double`	
`long double`	

Exercise 3: Debug stubs with printf

Navigate to the directory: **~/p1.2015s1/lab04/working_copy/ex03/**

Open **lab04ex03.c** and add the following code to the file. Compile and run the program.

```
#include <stdio.h>

int main(int argc, char* argv[])
{
    printf("a) Starting main function.\n");

    printf("b) About to start looping...\n");

    for (int i = 0; i < 8; i++)
    {
        // Add your code here...
    }

    printf("c) Finished looping...\n");

    printf("d) Ending main function.\n");

    return (0);
}
```

Add code inside the loop which cause the program to print out the following output:

```
a) Starting main function.
b) About to start looping...
...Inside loop, iteration: i is currently storing: 0
...Inside loop, iteration: i is currently storing: 1
...Inside loop, iteration: i is currently storing: 2
...Inside loop, iteration: i is currently storing: 3
...Inside loop, iteration: i is currently storing: 4
...Inside loop, iteration: i is currently storing: 5
...Inside loop, iteration: i is currently storing: 6
...Inside loop, iteration: i is currently storing: 7
c) Finished looping...
d) Ending main function.
```

Test your program to ensure it outputs the required text accurately, make sure the spaces you output are the same as the above sample output!

Once complete, have a TA review your completed exercises 1, 2 and 3 for this lab session. See the end of this document for the review questions.

Exercise 4: Using variables and printf

Navigate to the directory: **`~/p1.2015s1/lab04/working_copy/ex04/`**

Open **`lab04ex04.c`** and add the following code to the file. Replace the comments with appropriate code to achieve the desired action of the comment as described below.

```
#include <stdio.h>

int adder(int a, int b)
{
    // Output: "Adding a with b" using the contents of a and b.

    return (a + b);
}

int multiplier(int a, int b)
{
    // Output: "Multiplying a with b" using the contents of a and b.

    return (a * b);
}

int main(int argc, char* argv[])
{
    int first_number = 4;
    int second_number = 7;

    // Output first_number.
    // Output second_number.

    int adding_result = adder(first_number, second_number);

    // Output adding_result.

    int multiplying_result = multiplier(first_number, second_number);

    // Output multiplying_result.

    return (0);
}
```

Test your program to ensure it outputs the required text accurately, make sure the spaces you output are the same as the above sample output!

Exercise 5: Using variables, printf and scanf

Navigate to the directory: **~/p1.2015s1/lab04/working_copy/ex05/**

Copy the completed ex04 source code into ex05, and name the file **lab04ex05.c**. Open **lab04ex05.c** and change the code such that the program queries the user for two numbers, then calculates the addition and the multiplication of these two numbers, and outputs these to the screen. You will need to use **scanf**.

Ensure your program behaves as follows when run:

```
Welcome to the Raspberry Pi Adder and Multiplier!
-------------------------------------------------

Input a number: 5
Input another number: 6

--------------
Calculating...
--------------

5 plus 6 is 11

5 times 6 is 30

-------
Thanks!
```

Exercise 6: Using srand, time and rand

Navigate to the directory: **~/p1.2015s1/lab04/working_copy/ex06/**

Open **lab04ex06.c** and write code to create the following program. The program must roll two six-sided dice, and show the user the result. The user must then be queried as to whether they want to roll again. Ensure your program behaves similar to as follows when run:

```
Dice Roller:
------------

Rolling Dice 1: 5
Rolling Dice 2: 3
Roll again? Y

Rolling Dice 1: 4
Rolling Dice 2: 2
Roll again? N

You rolled the dice 2 times, goodbye!
```

Exercise 7: Character input and the Turtle World

Navigate to the directory: **`~/p1.2015s1/lab04/working_copy/ex07/`**

Open **`lab04ex07.c`** and write code to create the following program. When the program starts, it prints a menu:

```
Rock-Paper-Scissors:
--------------------

Type:
     R to choose rock
     P to choose paper
     S to choose scissors

What is your choice?
```

The player must then enter their choice. If they enter an unacceptable choice, the program should alert the user to the incorrect input and then query for their choice again.

Once the program has obtained a valid choice, it should generate a computer AI choice, and then launch the P1 Turtle World. The program must then draw the player's choice and the AI's choice, and then announce with turtle drawing whether the player wins or loses!

Be sure to include the required header files at the top of your program! You may need to consult the lecture materials.

One tested and complete, extend your program to be Rock-Paper-Scissors-Lizard-Spock; you may need to "google" the rules!

Week 04, Lab 04 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 04 Lab 04 Weight 1%	Yes	No
Ex 1:	Program accurately outputs required message?		
	File comment completed?		
Ex 2:	Program accurately outputs sizeof each type?		
	Table correctly complete on lab handout?		
Ex 3:	Loop iterations accurately output as required?		
Ex 4:	Comments replaced correctly with printf calls?		
Ex 5:	Program queries the user for two numbers?		
	User input numbers are added and result output?		
	User input numbers are multiplied and result output?		
Ex 6:	Two six-sided dice rolled?		
	User can choose to re-roll the dice?		
Ex 7:	Menu displayed at start of program?		
	User can select a move (rock or paper or scissors)?		
	AI move displayed using P1 Turtle World?		
	Winner announced in P1 Turtle World?		
	Total:		

Next activity: Final Week 4 Lecture and Week 3 and Week 4 Homeworks.