Steps to Create a Simple Linux Kernel Module
LAB3 - instruction

1. Set Up the Environment
Install build tools and kernel headers (on Ubuntu):

Note : to fixed any problem with update, run this:
sudo sed -i -re 's/([a-z]{2}\.)?archive.ubuntu.com|security.ubuntu.com/old-releases.ubuntu.com/g' /etc/apt/sources.list

sudo apt-get update
sudo apt-get install build-essential linux-headers-$(uname -r)

2. Create Source Code for the Module
Make a new folder (e.g. hello_module).
In a file called hello.c, put the following code:

```
#include <linux/init.h>
#include <linux/module.h>
#include <linux/kernel.h>

MODULE_LICENSE("GPL");
MODULE_AUTHOR("OSLAB404LAB3");

static int __init hello_init(void) {
   printk(KERN_INFO "Hello, World!\n");
   return 0;
}

static void __exit hello_exit(void) {
   printk(KERN_INFO "Goodbye, World!\n");
}

module_init(hello_init);
module_exit(hello_exit);
```

3. Create a Makefile
In the same folder, add a file named Makefile:
Note: be careful with spaces

```
obj-m += hello.o
all:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) modules
clean:
	make -C /lib/modules/$(shell uname -r)/build M=$(PWD) clean
```

4. Build the Kernel Module
Run in terminal:

```
make
```

5. Load and Unload the Kernel Module
Load the module:
Before loading the module you can list the current module on the terminal by:

```
lsmod
```

```
sudo insmod hello.ko
```

To be sure about loading the specific module, you can run in terminal:
lsmod grep *module_name*

Check message output:
Run in terminal:

```
dmesg
dmesg | tail
```

Unload the module:
Run in terminal:

```
sudo rmmod hello
```