Lab Exercise - Week 1

1. Getting ready for the lab

(If you see this page, you have completer Task1)

- Login to the Ubuntu/Linux machine
- Change the initial password
- Set up internet access (proxy)
- Access your institute email
- Access moodle - the course management system

2. Using the Linux terminal

Ubuntu Key (Dash) + Type terminal

Ctrl+Alt+t

3. Basic terminal commands

pwd - to know the current directory

mkdir - to create a new directory

(In linux, all commands and names are case sensitve.

cd and CD are not the same

File and file are note the same)

cd - to change directory

cd ../

cd

ls - listing all the files in the directory

ls -lh

4. Task 1 : Use the terminal to

Create a directory named CS1020 inside your home directory

```
cd
mkdir CS1020
```

The first command takes you to your home directory from where ever you are.
The second command creates a new directory named CS1020 in your home directory

Create a directory named Week01 inside CS1020

```
cd CS1020
mkdir Week01
```

The first command takes you inside the new directory CS1020
The second command creates a new directory named Week01 inside the CS1020 directory

5. Gedit - The editor to write all your programs

```
cd Week01
gedit &
```

The first command takes you inside the new directory Week01
The second command opens a text-editor program called "gedit".
The "&" is to make the gedit program run in the background so that the terminal is still usable for other commands.
You can type something and save it as firstfile.c

6. First C program (hello world)

Step 1. Create an new file "hello.c" using gedit

```
cd
cd CS1020
cd Week01
gedit hello.c &
```

and type the following in the file:

```
#include <stdio.h>
int main()
{
  printf("Hello world\n");
  return 0;
}
```

Step 2. Save the file hello.c

Step 3. Compile the program

In the terminal, using cd, go to the directory which contains the program and execute the command

```
gcc hello.c
```

If your program had no mistakes, gcc will silently finish its job. If it prints something on the screen, then you may have made some mistake in the program. Note that unlike humans, computers cannot read words with wrong spellings or sentences with wrong punctuation.

Step 4. Run the compiled output

```
./a.out
```

7. Program to print a integer

Program name: print_integer.c

Step 1. Create an new file "print_integer.c" using gedit and enter the following

```
#include <stdio.h>
int main()
{
  int x = 15;
  printf("%d\n", x);
  return 0;
}
```

Step 2. Save the file print_integer.c

Step 3. Compile the program

In the terminal, using cd, go to the directory which contains the program and execute the command

```
gcc print_integer.c
```

Step 4. Run the compiled output

```
./a.out
```

8. Program to print a real number (number with a decimal point)

Program name: print_realnumber.c

```
#include <stdio.h>
int main()
{
  float x = 1.414;
  printf("%f\n", x);
  return 0;
}
```

9. Program to read an integer (from the user) and print it back

This program has a line missing. Try to find it.

Program name: read_print_integer.c

```
int main()
{
  int x;
  printf("Enter an integer: ");
  scanf("%d", &x);
  printf("%d\n", x);
  return 0;
}
```

10. Program to read a real number (from the user) and print it back

Program name: read_print_realnumber.c

11. Homework Assignments

(to be completed before next lab session)

1. A program to read an integer and display its square
2. A program to read two integers and display their sum
3. A program to read a positve integer n and display the sum of numbers from 1 to n.

Good luck