TK1914 (C++ PROGRAMMING)
LAB 1 - INTRODUCTION TO LINUX (WITH UBUNTU)

INITIAL STEPS TO LOGIN INTO UBUNTU WORKSTATION

1. Boot into Ubuntu OS. Refer to your lab tutor on how to do this.

2. Login into the Ubuntu workstation using your supplied login and password.

3. Open Ubuntu's terminal client by any of the method below:
   - Right click on the desktop and select terminal icon; or
   - On the start menu, highlight application and select terminal icon; or
   - Find on the desktop window if there's an icon called terminal or shell.

UBUNTU TERMINAL CLIENT

1. Linux commands are usually written through Ubuntu's terminal.

2. Previously Linux heavily depends on terminal clients (command lines) to execute operations, compared to Microsoft Windows which is fully GUI based.

3. Now, most Linux OS comes with rich set of GUI features which is on par with MS Windows, but most operations still depend on linux command lines.

ACCESSING UBUNTU SERVER FROM OUTSIDE LAB USING TELNET

1. If you're outside the Ubuntu Lab and still want access to Ubuntu server, you can do so by using telnet.

2. Telnet enables you to control the server and communicate with other servers on the network.

3. To access from outside the lab, follow the steps below:

   - On windows command line, type the following:

     ```
     telnet 202.185.46.9
     ```

   - If connection is available, then it will ask for a login and password. Enter yours. For example:

     ```
     Login: a1234567
     Password: 1234567
     ```

BASIC LINUX COMMANDS

Enter the following commands in the terminal client window. Observe the result of each command:

```
/> pwd                          (to find the name of the working directory)

/> ls                           (to list the files and directories)

/> mkdir MyDirectory            (to make a directory called MyDirectory)

/> cd MyDirectory               (to change working directory to MyDirectory)

/> cd /                         (change directory to root)

/> cd ~                         (to change to user home directory)

/> cd ..                      (to change to a directory above)

/> pico test1.cpp               (to create a new file called test1.cpp)
```

Inside the pico text editor, type the following code below:

```
    #include <iostream>
    #include <string>       // header file for string variable
    using namespace std;

    int main()
        {
        string name;
        cout << “Please enter your name” << endl;
        cin >> name;
        cout << “Your name is “ << name << endl;
        return 0;
        }
```

CTRL + O : Save File
CTRL + X : Exit
CTRL + Y : Go to first cursor (top)
CTRL + V : Go to last cursor (bottom)

Save the file.

/> cat test1.cpp	(to display the content of test1.cpp)
/> more test1.cpp	(another way to display the content of test1.cpp)
/> less test1.cpp	(yet another way to display content of test1.cpp)

<u>Commands to copy and paste files and directory</u>

/> cp test1.cpp test2.cpp	(to copy test1.cpp into test2.cpp)
/> cp test1.cpp MyDirectory/	(to copy file test1.cpp into directory MyDirectory)
/> cp –r MyDirectory MyDirectory2	(to copy directory MyDirectory and paste it as directory MyDirectory2)

<u>Commands to cut and paste files and directory</u>

/> mv test1.cpp test3.cpp	(to cut file test1.cpp and paste it as file test3.cpp)
/> mv test3.cpp MyDirectory/	(to cut / paste file test1.cpp into directory MyDirectory)

<u>Commands to delete files and directory</u>

/> rm test2.cpp	(to remove file test2.cpp)
/> rm –r MyDirectory2	(to remove directory MyDirectory2)

<u>Commands to get more info on specific Linux commands</u>

/> man ls	(manual / help description on linux command ls)
/> man rm	(manual / help on linux command rm)

C++ COMPILATION STEPS IN LINUX (WITH G++)

/> cd MyDirectory	(go to working directory where file test1.cpp is located)

Compiling the file test2.cpp (method 1)

/> g++ test1.cpp	(to compile file test1.cpp)

Now to run that compiled file (method 1)

/> ./a.out	(by default, to run the compiled program from test2.cpp)

Compiling the file test2.cpp (method 2)

/> g++ test2.cpp –o test2.out	(another way to compile file test2.cpp, but directly specify

output as test2.out)

Now to run that compiled file (method 2)

```
/>  ./test2.out                    (to run the compiled program which has been specifically
                                   named test2.out)
```

ALTERNATIVE TEXT EDITOR

Below are some of other available text editors in linux, try them by typing below on the terminal client:

```
/>  gedit test3.cpp

/>  vi test3.cpp

/> emacs test3.cpp
```

1. Bear in mind that there are many text editors developed in Linux. Some are text based / command based like vim, others are just like microsoft windows notepad (e.g. emacs).

2. Note that each text editor in linux also has its own command keys, just search in google to find out more on how to use these text editors.
   For example, on google, type: “linux vim tutorial”

*End of Lab 1.