



**Exercise Manual
for
CE/CZ1003
Introduction to Computational Thinking**

**Practical Exercise #1:
Introduction to Raspberry Pi**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY**

Ex. #1 - Introduction to RPi

Learning Objectives

In this course, students will learn the various concepts of computational thinking through programming exercises performed on the Raspberry Pi platform. This practical exercise is to introduce students to the Raspberry Pi (RPi) single-board computer, and how to use its text-based (Linux's bash) commands to explore the environment of the RPi.

Intended Learning Outcomes

At the end of this exercise, you should be able to

- Operate the Raspberry Pi board through its Graphic User Interface and through its Terminal mode interface.
- Write basic programs in Python language and C language to observe the difference in running an interpreted language and a compiled language based programs.

Equipment and accessories required

- i) Raspberry Pi 3 Model B (RPi3) board with Sense HAT add-on display module/board.
- ii) HDMI monitor, USB keyboard and USB mouse.
- iii) A USB power source to power the RPi3 board (E.g. Power Bank, Adaptor or USB port of a desktop computer).

1. Introduction

The Raspberry Pi (RPi) is a small single-board computer developed in UK by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools. It is a low cost computer board (about S\$50) but comes with powerful processor (that are also used in some of the latest smartphones).



Fig.1 - Raspberry Pi 3 Model B (RPi3)

For the RPi3 model that is used in this exercise, the board also support two wireless interfaces (i.e. connections), Wifi and Bluetooth, which means that it can also be accessed using your wireless devices such as Notebook and Smartphone.

Ex. #1 - Introduction to RPi

As an educational kit, you can add other devices to the RPi board, typically in the form of add-on modules. In this course, you will be using the **Sense Hat** add-on module on the RPi board, such that you can code programs to display messages and images on the module, as will be seen later.

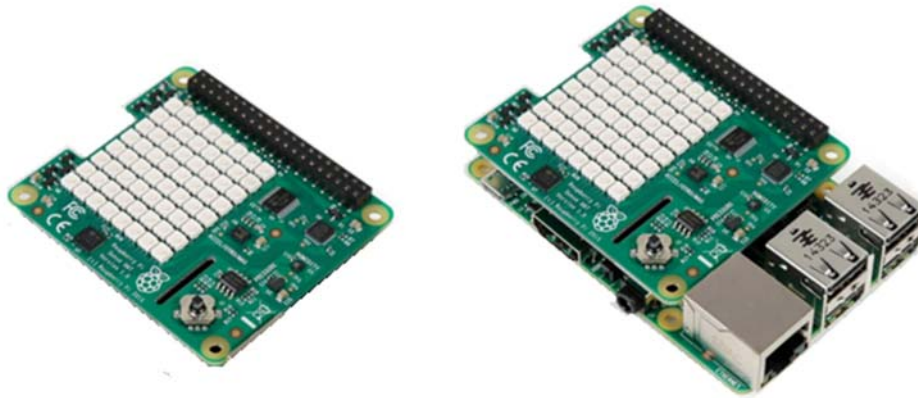


Fig. 2 - Sense Hat module and its interface on RPi3

The RPi can be set up as a standalone computer by connecting a monitor, keyboard and mouse through its various interfaces.

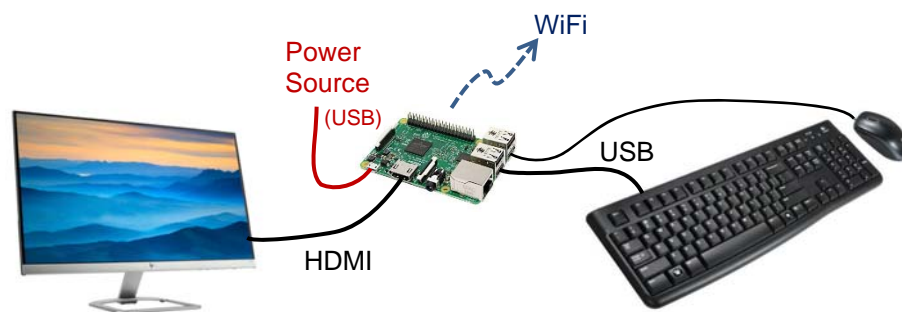


Fig. 3 - Standalone setup of RPi3

The operating system (OS) used on the RPi is the Raspbian, which is based on the open source Linux OS. (An OS is the software used to control and operate a computer, such as the Microsoft's Windows OS and Apple's macOS for their notebooks, and Google's Android and Apple's iOS for their smartphones)

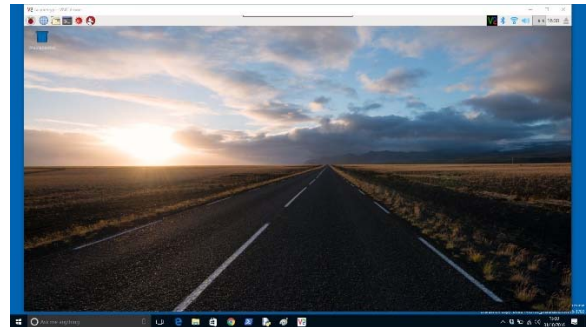
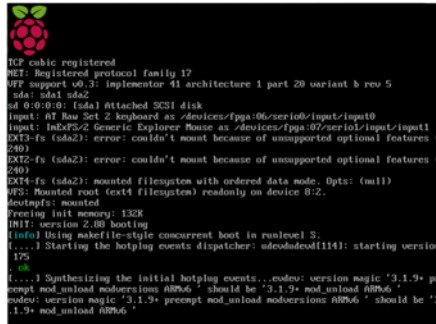
2 Familiarization of the RPi's User Interface and useful commands

You will now go through a sequence of tasks designed to get you familiarized with the operation of the RPi as a standalone computer, in particular, certain commands that are frequently used on RPi.

- 2.1 Connect the RPi3 board with the keyboard, mouse and monitor as shown in Figure 3.
- 2.2 Apply the power source to the RPi board (which in this case is by connecting the USB cable to one of the USB port desktop PC in the lab)

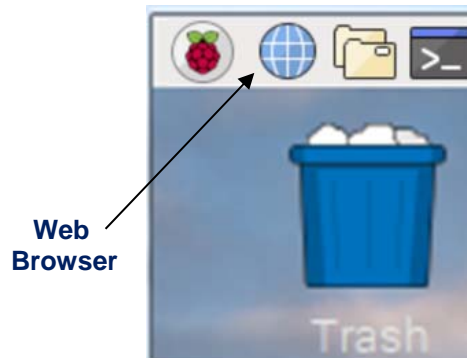
Ex. #1 - Introduction to RPi

- You will see a series of messages appearing on the monitor, which eventually leads to a graphic based display screen – the Desktop screen.



2.3 This is the Graphic User Interface (GUI) based Desktop provided by Raspbian. There are several 'icons' available on the Desktop that allow you to interact with the RPi3 system.

- The icons on the top left corner of the Desktop screen can be used to access the various functions available on RPi, including a Web browser program that you are going to use in a while.

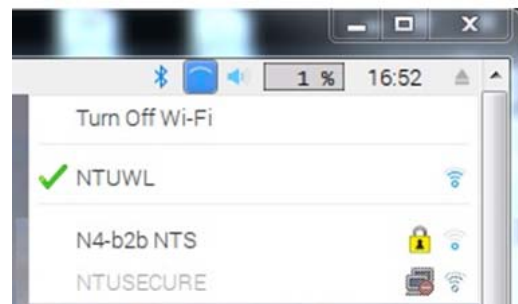


2.4 To start with, you are going to connect the RPi3 to NTU wireless network, and then log in to the NTU Learn's website to access the various relevant documents needed for doing the Cx1003 hands-on exercises – including this exercise manual.

- On the top right corner of the Desktop, click the Wi-Fi icon, and there should be the 'NTUWL' connection that you can select.

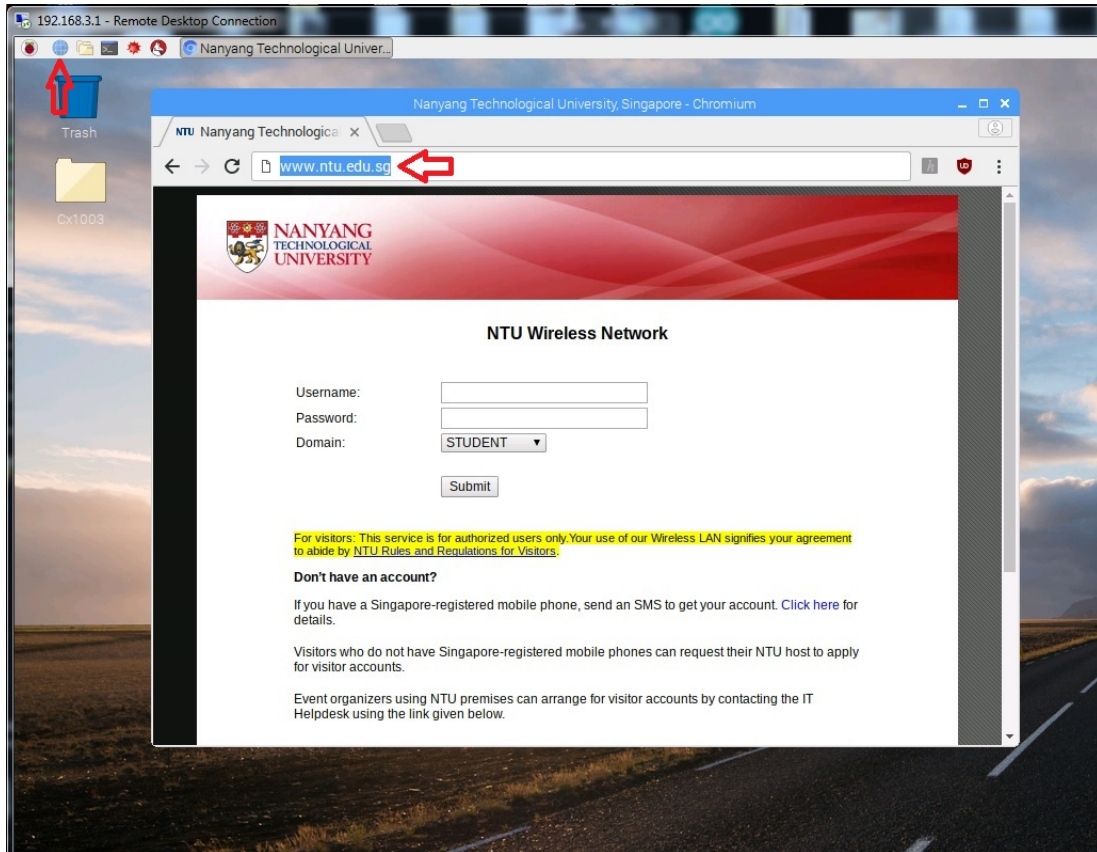


Wi-Fi



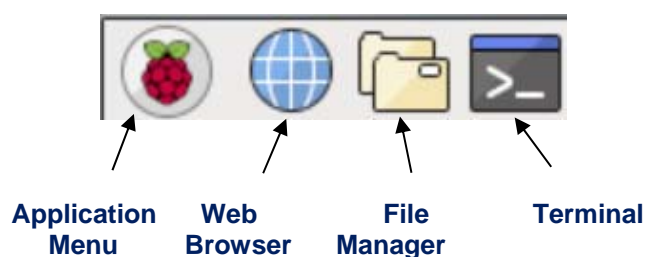
Ex. #1 - Introduction to RPi

- Next, launch the web browser program at the top left corner of the Desktop. Enter 'www.ntu.edu.sg' in the URL address bar, and the NTU network login interface should appear.



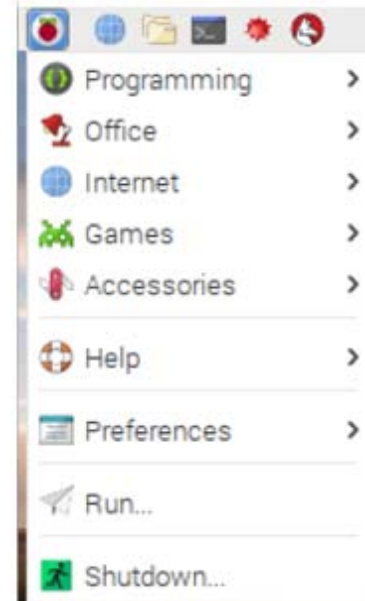
- Enter your NTU account's detail, and if everything is OK, the RPi board should now be connected to the NTU's network.
- You can now log on to the NTULearn website, 'ntulearn.ntu.edu.sg' to retrieve this exercise manual, and continue with the rest of the hands-on exercises.

2.5 Back to the icons on the top left corner of the Desktop screen, these icons can be used to access the various functions available on RPi.

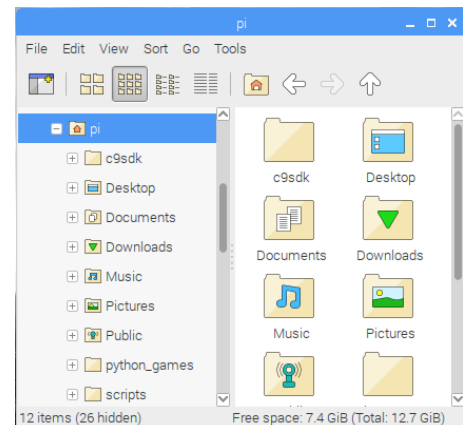


Ex. #1 - Introduction to RPi

- Click on the **Application Menu** icon - you will be provided with a list of options - programs and functions available on RPi, which you can use to operate the RPi. Note that there is a **Shutdown** function which is used to properly turn off the RPi at the end of the exercise.
- The **Web Browser** is normally used for surfing internet, which you will use as needed during the various hands-on exercises in this course.
- File Manager** is used to explore the directories and files stores on the RPi.
- Terminal** is a program that is usually used to enter text commands and run other programs, as will be seen later.



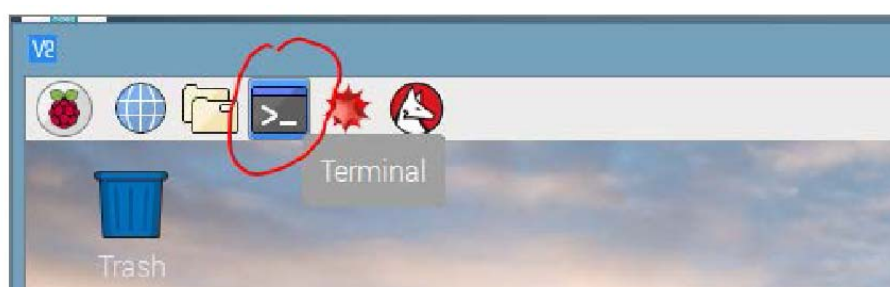
- 2.6 Click on the **File Manager** icon. You will be presented with a window showing the various folders/directories and files available on the RPi.
- You can further explore the various folders by clicking on each of them to see their contents.
 - Click the 'Desktop' folder and explore its content.



3. Finding information using commands in Terminal mode

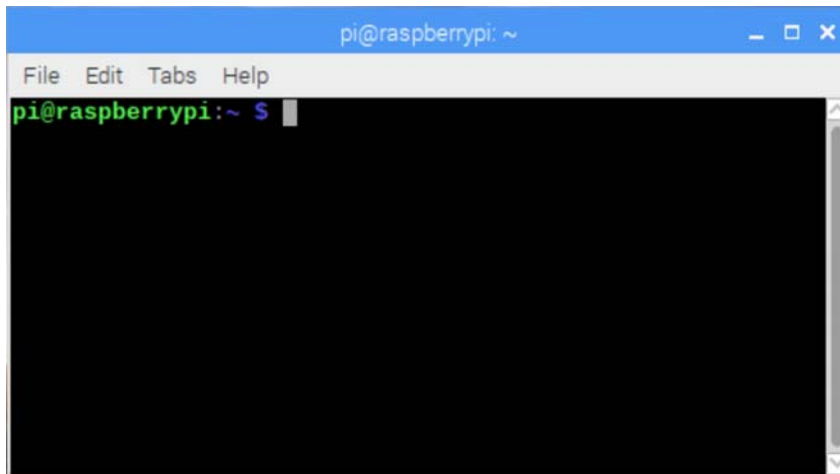
While the GUI based Desktop is convenient and user friendly, there are other things that you can do with RPi, many of which can only be accessed by typing text-based commands through the Terminal. In practice, using Terminal mode is the most 'powerful', flexible and quickest way to accomplish many things on RPi.

- 3.1 Click on the **Terminal** icon to open a terminal window.



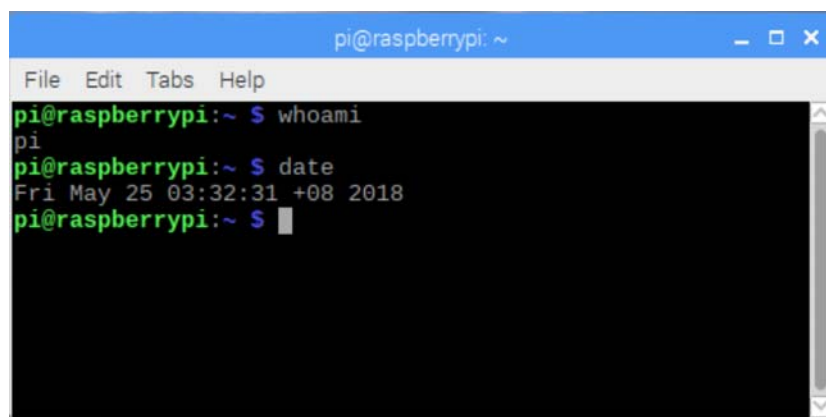
Ex. #1 - Introduction to RPi

- Resize the terminal window so it's of a reasonable size for you to type into.

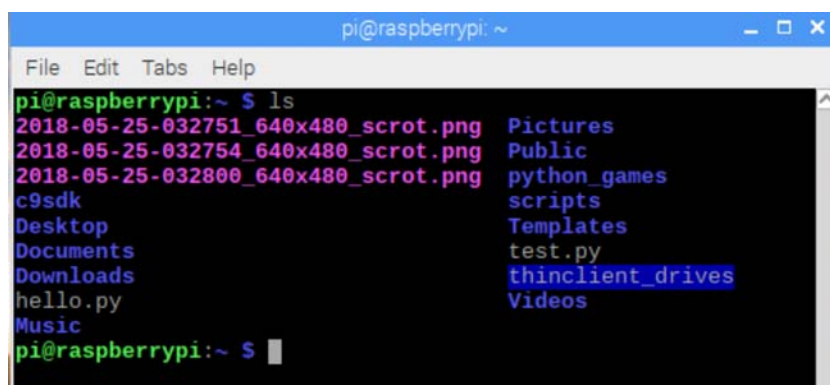


3.2 Type the following commands into the terminal and observe what is shown for each of them.

- **whoami**
- **date**



3.3 You can view the contents in the current directory by using the list command **ls**. This will show the various files and sub-directories available in the current directory.



Ex. #1 - Introduction to RPi

- 3.4 You can change to a different directory, such as to the **scripts** directory using the command: **cd scripts**

```
pi@raspberrypi:~ $ cd scripts/  
pi@raspberrypi:~/scripts $ ls  
clean.sh  cloud9_startup.sh  
pi@raspberrypi:~/scripts $
```

- View the contents of this directory (using the **ls** command again).

- 3.5 You can view the content of the file by using the command **more**

- Type the command **more cloud9_startup.sh**
- The content of this file is then shown on the screen.

```
pi@raspberrypi:~/scripts $ more cloud9_startup.sh  
cd ~/c9sdk  
/usr/bin/nodejs server.js -p 8181 -l 0.0.0.0 -a pi:raspberry -w /home/pi/Desktop/Cx1003  
pi@raspberrypi:~/scripts $
```

Helpful tip: The **<Tab>** key is very useful when typing command in the terminal mode, such as to select a file in a directory without entering the full name of the file. E.g., you can execute the previous command by typing **more c<Tab>**. Try it.

- 3.6 You can change to the directory above the present directory as follows

- Type the command **cd ..**

There are many other commands and things that you can do in terminal mode, which you will learn in due course, but we will now move on to do a bit of programming exercises on the RPi board.

4. Interpreted language versus Compiled language

You have learnt in Topic 1 of the recorded video about the different type of programming languages, which can be further separated into interpreted language, such as Python and compiled language such as C. In this exercise you are going to write a simple program using each of these two languages and observe how each of them get executed.

- 4.1 Python is an 'interpreted language'. You write the code and run the program. There are two ways to run a python code, the Script mode and the Interactive mode. You will use the Interactive mode here which is clearer for illustrating the working of an interpreter.

- Open a terminal window in RPi and launch the Python **interpreter, python3** by entering

python3

- You will see the prompt **>>>** display on screen

```
pi@raspberrypi:~ $ python3  
Python 3.5.3 (default, Jan 19 2017, 14:11:04)  
[GCC 6.3.0 20170124] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>
```


Ex. #1 - Introduction to RPi

- Key in the following Python code, and press *Enter*

```
print ("Hello world")
```

You will see that the message is shown on the screen immediately in the next line.

```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Hello world")
Hello world
>>>
```

And that is – the code that you entered has been executed by the interpreter python3.

- Next key in the following and press *Enter*

```
1+2
```

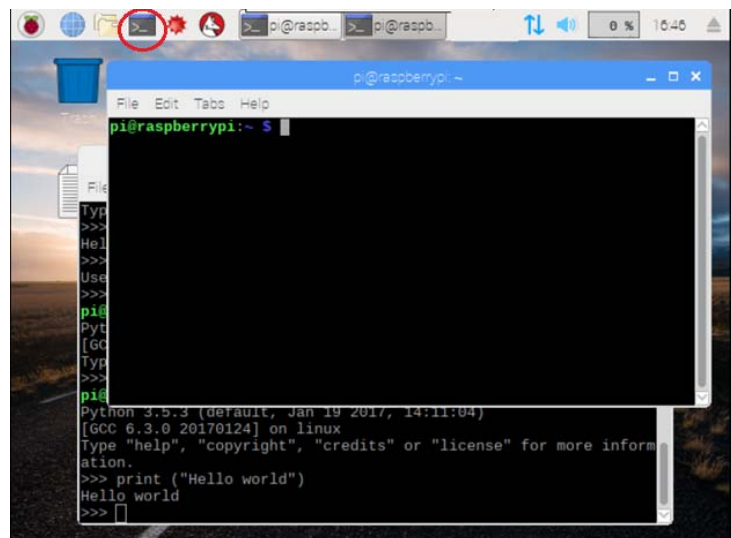
You will notice that the code is immediately executed by the interpreter as soon as you press the *Enter* key.

```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Hello world")
Hello world
>>> 1+2
3
>>>
```

(In practice, we will usually use the Script mode as described in Appendix A, which is a more practical way to develop a python program)

- Let us now write a program in C to do exactly the same thing. Unlike the Python program, you need to first create a file to key in the program code in C. In here we will use a text editor program called **nano**.

- To do so, you can either use the existing Terminal (i.e. the one still runs the Python3 interpreter), or you can open another terminal by clicking the **Terminal** icon.
- We will open another Terminal here. (This is to also demonstrate the multi-tasking ability of the RPi, i.e. its ability to run multiple programs at the same time).



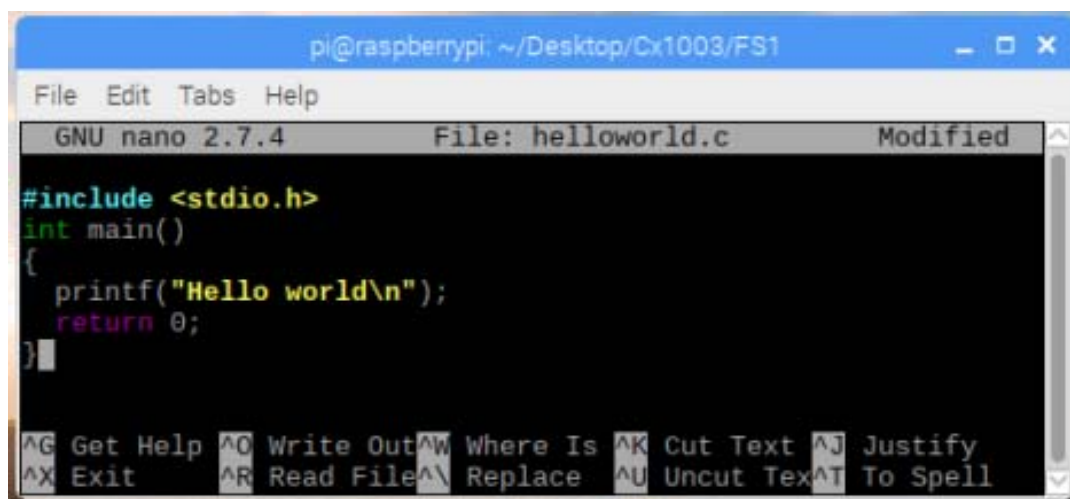
Ex. #1 - Introduction to RPi

- First, change to the subdirectory of your lab group (e.g. FS1 in the screenshot below). Then use the text editor program, **nano** to create a file calls **helloworld.c** using the command as shown below



```
pi@raspberrypi: ~/Desktop/Cx1003/FS1
File Edit Tabs Help
pi@raspberrypi:~ $ ls
c9sdk  Documents  Music      Public      scripts  thinclient_drives
Desktop Downloads Pictures  python_games  Templates Videos
pi@raspberrypi:~ $ cd Desktop/Cx1003/FS1
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ nano helloworld.c
```

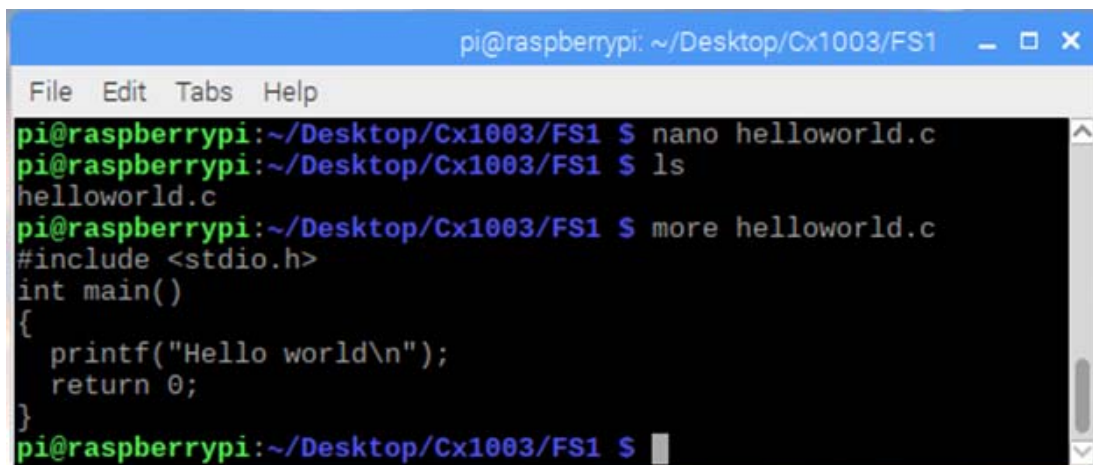
- Key in the program code as shown below



```
pi@raspberrypi: ~/Desktop/Cx1003/FS1
File Edit Tabs Help
GNU nano 2.7.4      File: helloworld.c      Modified
#include <stdio.h>
int main()
{
    printf("Hello world\n");
    return 0;
}
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Tex ^T To Spell

- Save the program and close the nano text editor by pressing the keys Ctrl-X (i.e. *Ctrl* key together with the *X* key).
- What you have done so far is you have created a C program file. You can check the existence of this file and see its contents using the commands that you have learnt earlier.



```
pi@raspberrypi: ~/Desktop/Cx1003/FS1
File Edit Tabs Help
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ nano helloworld.c
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ ls
helloworld.c
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ more helloworld.c
#include <stdio.h>
int main()
{
    printf("Hello world\n");
    return 0;
}
pi@raspberrypi:~/Desktop/Cx1003/FS1 $
```

Ex. #1 - Introduction to RPi

- In order to execute this C program, you will need to first compile it. To do so, you will need to run a compiler, and here you will use the **gcc compiler**. Type the gcc command as shown below. If everything is OK, then a new file **hello** will be created as shown - this is the executable file created by the compiler.

```
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ gcc -o hello helloworld.c
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ ls
hello helloworld.c
pi@raspberrypi:~/Desktop/Cx1003/FS1 $
```

- If instead, you see other message(s) and cannot find the hello file, it means that you probably have made a mistake somewhere, either in your code or in the command that you typed. This is then a great opportunity for you to learn about 'debugging' – to find out what is/are wrong and fix the problem(s).
- Once you have successfully generate the executable program, type the command as shown below to run it

./hello

```
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ gcc -o hello helloworld.c
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ ls
hello helloworld.c
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ ./hello
Hello world
pi@raspberrypi:~/Desktop/Cx1003/FS1 $
```

- If everything is OK, then the message will be printed on screen to show the output of your program.

At this juncture, you should review what you had done in these two simple programming exercises. Make sure that you understand the various steps you used to create and compile the C program to produce an executable file. (There are also more exercises in the Appendix for you to try if time allows.)

5. Displaying message on Sense Hat Module

Instead of displaying message on screen, you can also output message to the Sense Hat module with the RPi-Sense Hat setup. To provide a preview of how the Sense Hat module will be used in subsequent hands-on exercises, you will now code a Python program to display a message on the Sense-Hat module.

- First select the python3 Terminal that you opened earlier on.
- Type the following **3 lines of code** in the Terminal, and observe the display on the Sense HAT module.

```
>>>from sense_hat import SenseHat
>>>sense = SenseHat()
>>>sense.show_message("Hello NTU")
```

- You can terminate the running program by pressing the two keys **Ctrl+C**

Ex. #1 - Introduction to RPi

6. Summary

In the hands-on exercises here, you have learnt how the RPi can be setup as a standalone computer. You have also learnt some of the basic commands used to navigate the RPi system. You have tried writing two basic programs, in Python language and C language to appreciate the difference in interpreted language and compiled language.

Lastly, issue the following command in the Terminal to properly shut down the RPi computer (and goggle to find out what the command means).

sudo shutdown -h 0

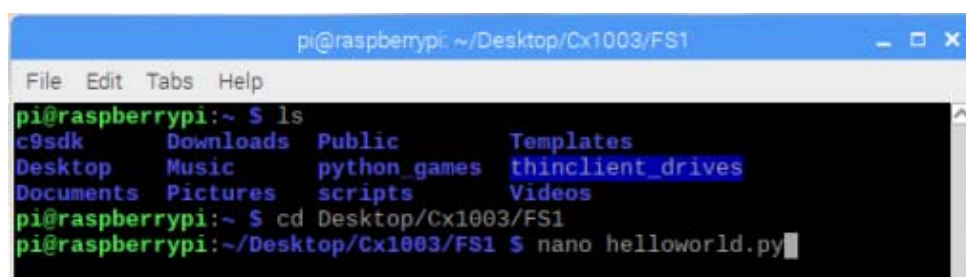
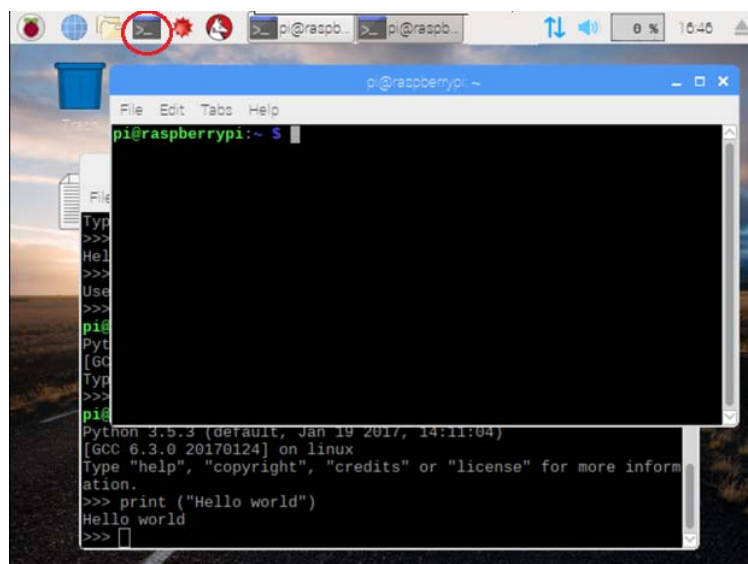
Appendix

You may want to try the following two extra exercises if time permits.

A. Python Script Mode

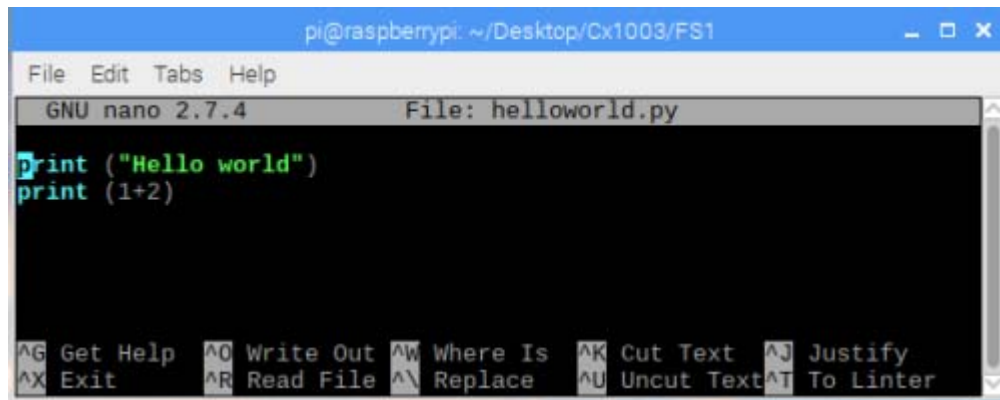
When we write python program for real-world application, we will usually store the python program code in a file, also known as script. We then run the interpreter to execute the script file.

- Open another Terminal, and change directory to the subdirectory of your lab group (e.g. FS1).



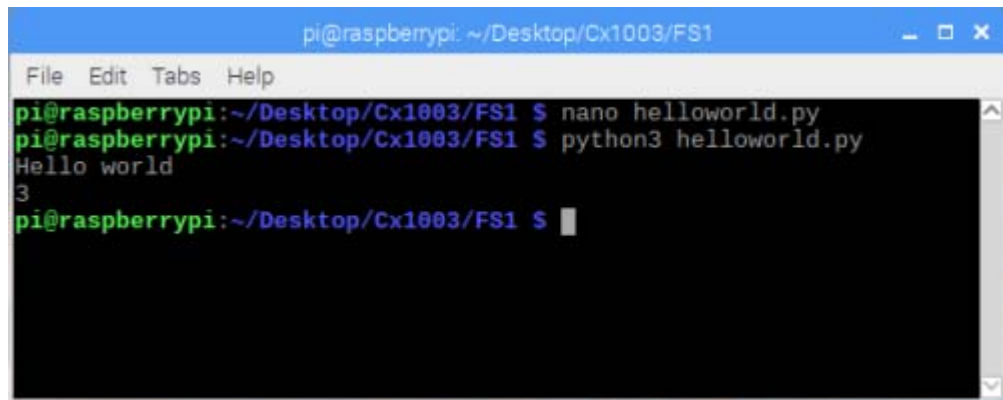
Ex. #1 - Introduction to RPi

- Use the text editor program, **nano**, to create a file calls **helloworld.py**.
- Key in the code and save the file by pressing the keys Ctrl and X. This is the file that contains the program with code written in Python.



```
pi@raspberrypi: ~/Desktop/Cx1003/FS1
File Edit Tabs Help
GNU nano 2.7.4 File: helloworld.py
print ("Hello world")
print (1+2)
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter
```

- You can now use the python3 interpreter to run the python program, and you should see the corresponding messages shown on the screen.

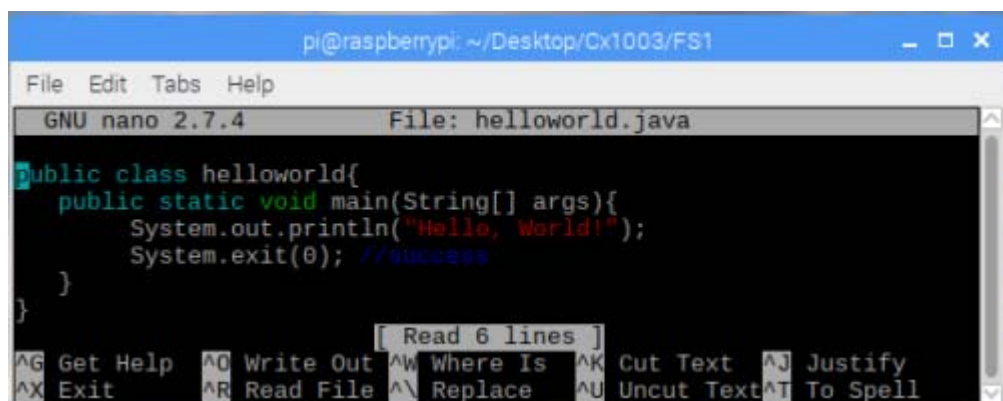


```
pi@raspberrypi: ~/Desktop/Cx1003/FS1
File Edit Tabs Help
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ nano helloworld.py
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ python3 helloworld.py
Hello world
3
pi@raspberrypi:~/Desktop/Cx1003/FS1 $
```

B. Basic Java Program

In contrast to Python and C, Java uses both a compiler and an interpreter to run its program. The following exercise shows the basic steps to develop and run a basic Java program.

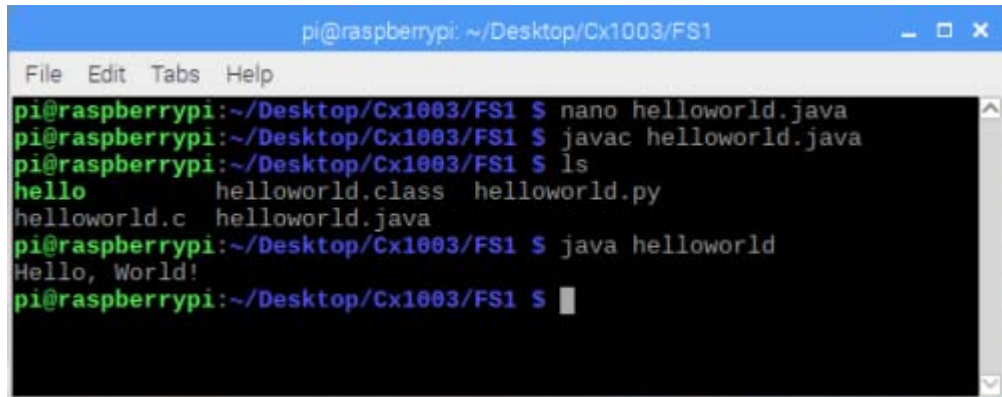
- Use nano to create a text file *helloworld.java* and key in the code as shown below.



```
pi@raspberrypi: ~/Desktop/Cx1003/FS1
File Edit Tabs Help
GNU nano 2.7.4 File: helloworld.java
public class helloworld{
    public static void main(String[] args){
        System.out.println("Hello, World!");
        System.exit(0); //success
    }
}
[ Read 6 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell
```


Ex. #1 - Introduction to RPi

- Next compile the java program by using the *javac* compiler



```
pi@raspberrypi: ~/Desktop/Cx1003/FS1
File Edit Tabs Help
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ nano helloworld.java
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ javac helloworld.java
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ ls
hello          helloworld.class  helloworld.py
helloworld.c  helloworld.java
pi@raspberrypi:~/Desktop/Cx1003/FS1 $ java helloworld
Hello, World!
pi@raspberrypi:~/Desktop/Cx1003/FS1 $
```

- A java bytecode file ***helloworld.class*** is created if there is no error.
- The bytecode file can then be executed by the java interpreter using the following command

java helloworld

The corresponding message will then be displayed on screen.

Note that two programs are used in the process for this java program:

- **javac** the compiler
- and
- **java** the interpreter.