NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Laboratory Manual
for
CE/CZ1003
Introduction to Computational Thinking

Practical Exercise #6:
Procedural Abstraction
(Function and Module)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
NANYANG TECHNOLOGICAL UNIVERSITY

Ex. #6 – Procedural Abstraction

Learning Objectives

The manual provides information and exercises to let you apply the concept of procedural abstraction in the form of function for easy re-use, and further extend it to a module to make the function sharable by multiple programs.

Intended Learning Outcomes

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

- write a commonly use block of code in the form of function and module

Equipment and accessories required

i) Raspberry Pi 3 Model B (RPi3) board with Sense HAT add-on display module/board.
ii) A USB power source to power the RPi3 board (E.g. Power Bank, Adaptor or USB port of a desktop computer).
iii) A computer (desktop PC or notebook) with Ethernet port and cable for remote access of RPi3. Software (open source) to be installed on the computer – PuTTY, VNC Viewer and WinSCP

---

1. Procedural Abstraction - Function and Module

When you need to perform an operation multiple times in a program, it is more efficient to write that block of code in the form of a function which you can re-use by calling the function at the appropriate juncture in the program. The function can be further made into a module such that it can be shared by multiple programs using the **`import`** statement.

In this exercise, you will learn how to write a function, and then make it into a module that can be imported into your program.

2. Function

In earlier exercise 4, you coded a program that prompts the user to input the values of the three primary colours (red, green and blue) to be used to display a message. In this exercise, you will code a function **`get_color()`** that can be re-used in the program as shown below.

```
from sense_hat import SenseHat
sense = SenseHat()
sense.set_rotation(180)

#--- function get_color() --------------------------
def get_color(color):
    keep_looping = True
    no_of_try=1
    while keep_looping:
        color_str=input("Enter the value of the " + color + \
                        " color for message (0 to 255):")
            :

#--------------------------------------------------
r_int = get_color("red")
g_int = get_color("green")
b_int = get_color("blue")
msg_color = (r_int, g_int, b_int)
sense.show_message("I got it!", text_colour=msg_color)
```

Coding Exercise 6a

a) Write a function **`get_color(color)`** that takes a string parameter, "color" as the input and return the integer value of the color entered by the user, based on the code snippet given on page 2.
   - The function checks for valid value entered by the user, in the range from 0 to 255.
   - The function returns the valid value entered by the user
   - If the user does not enter a valid value after 3 tries, the function will return a default value of 0

3. Module

The function can be made into a sharable module such that it can be imported into a program as follows.

```
from sense_hat import SenseHat
from textcolor import get_color

sense = SenseHat()
sense.set_rotation(180)
#-----------------------------------------------------------
r_int = get_color("red")
g_int = get_color("green")
b_int = get_color("blue")
msg_color = (r_int, g_int, b_int)
sense.show_message("I got it!", text_colour=msg_color)
```

Coding Exercise 6b

- Using the function created in exercise 6a, make it into a module such that it can be imported into a program as shown above.