****

Laboratory Manual

for

SC1003

Introduction to Computational Thinking and Programming

Practical Exercise #5:

Procedural Abstraction

(Function and Module)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

**NANYANG TECHNOLOGICAL UNIVERSITY**

# 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

1. Raspberry Pi 3 Model B (RPi3) board with Sense HAT add-on display module/board.
2. A USB power source to power the RPi3 board (E.g. Power Bank, Adaptor or USB port of a desktop computer).
3. 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 3, 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 5a***

1. 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 shown below.

**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 5b***

* Using the function created in exercise 5a, make it into a module such that it can be imported into a program as shown above.