Lesson 3

Admin Matters

Programming Quiz

- Begins promptly at the start of **Session 3**
- You have 15 minutes. (If you are late you have less time)
- **NOTE: We will ONLY accept submissions via Vocareum.**

Raspberry Pi Sets

- Read the 1D mini project for week 3 and start assembling it

1D Project Proposal

- Begin thinking about what you want to do
- You can bounce your ideas off us before the presentation

Homework

- Homework problems continue and submission is on Vocareum.

Digital World + Chemistry Combined Assignment

- Please access the wikispaces page.
- Solve the problems progressively
- You can work in groups but you have to submit individually.
- Sorry about the Vocareum problems …

Pre-Reading

- Read **Week 4** materials before Session 1. I encourage you to post your queries on Piazza.

Recall Last Week's Lesson

1. Consider the following code snippet.

```
17 def multiply_by_two(a):
18
19     b = 2*a
20     print(b)
21     return b
22
23 b = 10
24 c = multiply_by_two(b)
25 print(b)
```

   a. What is printed on the screen when **line 20** is executed?

   b. What is printed on the screen when **line 25** is executed?

2. Consider the following code snippet.

```
 8
 9 def standing_wave(A,k,x,w,t):
10
11     y = A*sin(k*x)*sin(w*t)
12     return y
13
14 y = standing_wave(5,1,10,0.1,100)
15 print(y)
```

   a. The code will be executed without error. (T/F)

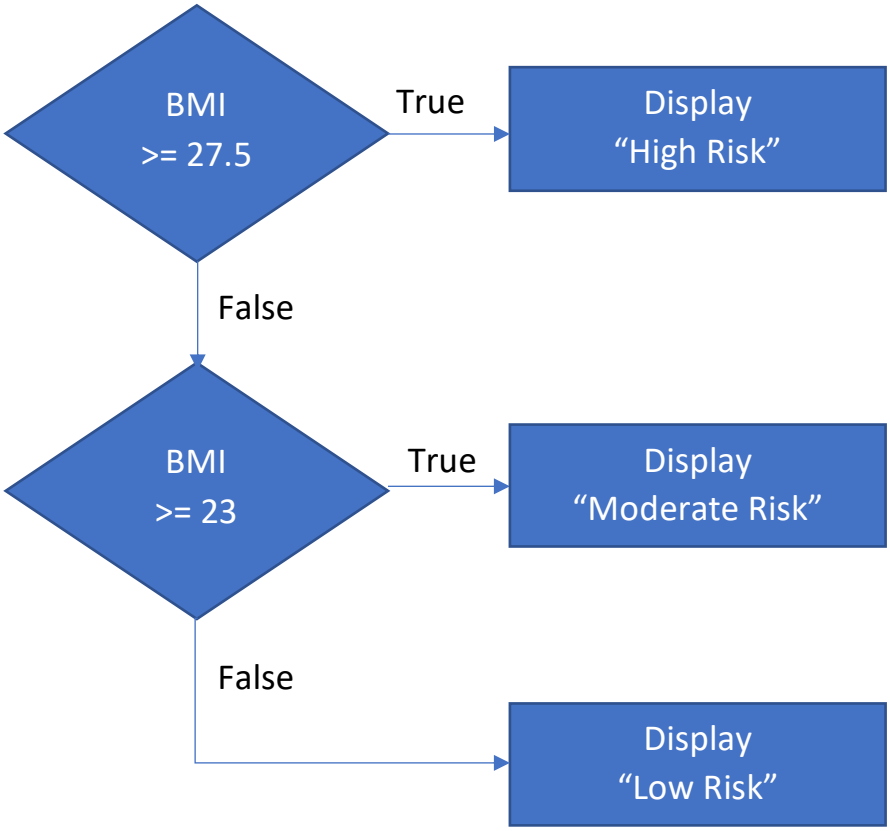
   b. The **name of the function** is `standing_wave`. (T/F).

   c. The **function call** is at line 14. (T/F)

   d. The **function header** is at line 9. (T/F)

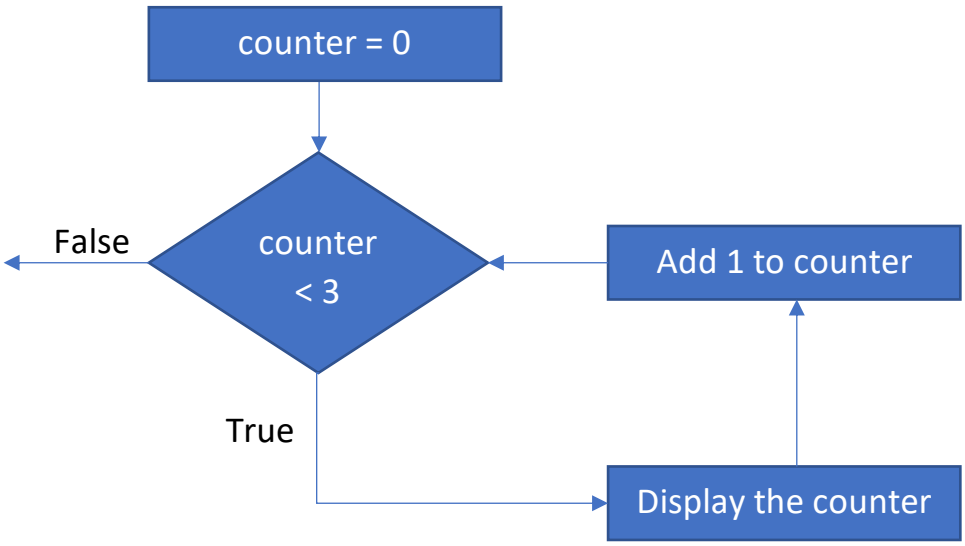
   e. The variable y at line 11 and line 14 are the same variable. (T/F)

Write the following using an if/elif/else structure



While loop

Remember this flowchart? Write this as a while loop.



General form

```
set condition to be true

while (condition):

    statements
    (one statement that makes condition false at some
    point in time)
```

The statements are executed as long as the condition remains true.

You need to ensure that the condition is true before beginning the loop.

Hence, you must have some way of making the condition false.

Using a counter

```
85 counter = 1
86 while( counter < 10 ):
87     print("ba")
88     counter += 1
```

The statement that somehow makes the condition false is at line _____.

Using the results of your calculation

Example. Let's say you want to calculate **total = 1 + 2 + 3 + …. + (n-1) + n** what must is the smallest value of n so that the total just exceeds 10000? Solve n(n+1)/2 = 10000 or use programming.

```
91 n = 0
92 total = 0
93 while(total < 10000):
94     total = total + n
95     n = n + 1
```

The statement that somehow makes the condition false is at line _____.

Use an if/else statement

```
condition = True
counter = 0
while(condition):
     if( counter == 300):
          condition = False
     counter += 1
```

How would you modify the code above to make an infinite loop?

Introduction to for-loops

You would like the following set of statements to help you with your currency conversion in your next trip.

```
"100 JPY is 1.19 SGD”
“200 JPY is 2.38 SGD”
“300 JPY is 3.57 SGD”
```

With what you know, you can write the following statements.

```
yen = 100
sgd = 1.19
print("{0} JPY is {1} SGD".format(yen, sgd)
print("{0} JPY is {1} SGD".format(2*yen, 2*sgd)
print("{0} JPY is {1} SGD".format(3*yen, 3*sgd)
```

If you are going to print up to 2000 yen, would you like to have 20 lines?
You can improve your productivity by using a for-loop with the range function.

```
yen = 100
sgd = 1.19
for i in range(1,21):
     print("{0} JPY is {1} SGD".format(i*yen, i*sgd)
```

`range(1,21)` gives you a sequence of integers beginning with 1 and ending with **20**. (It’s actually more complicated, but this is good enough for now).

`for i in range(1,21)` thus assigns the integers given by the range function sequentially to i at each iteration

Lists

For-loop. Looping over elements of a list individually

There are two methods that use a for-loop.
Which method gives you the **list element** at each iteration?
Which method gives you the **list index** at each iteration?

```
 97 #Method A
 98 my_list = [7, 1, 5, 9 , 0]
 99 for i in my_list:
100     print("Element is", i)
101
102 #Method B
103 length = len(my_list)
104 for i in range(length):
105     print("Element is", my_list[i])
```

The assignment statement copies the address of the list

```
108 my_list = [7, 1, 5, 9 , 0]
109 next_list = my_list
110 next_list[2] = "sutd"
111
112 print(my_list[2], next_list[2])
113 print(my_list is next_list)
```

Question. Line 113 gives True or False?
Use the `id` function or the variable explorer in spyder to see this clearly

To copy entries of a list, use list slices

Doing this is what is known as a shallow copy (as compared to deep copy).
More next week. Here we have three ways to do a shallow copy.

```
116 my_list = [7, 1, 5, 9 , 0]
117 a_list = my_list[:]
118 b_list = list(my_list)
119 c_list = my_list.copy()
```

More with lists (explore yourself)

(a) I need a list containing ten elements, all None or 0. How would you do it?

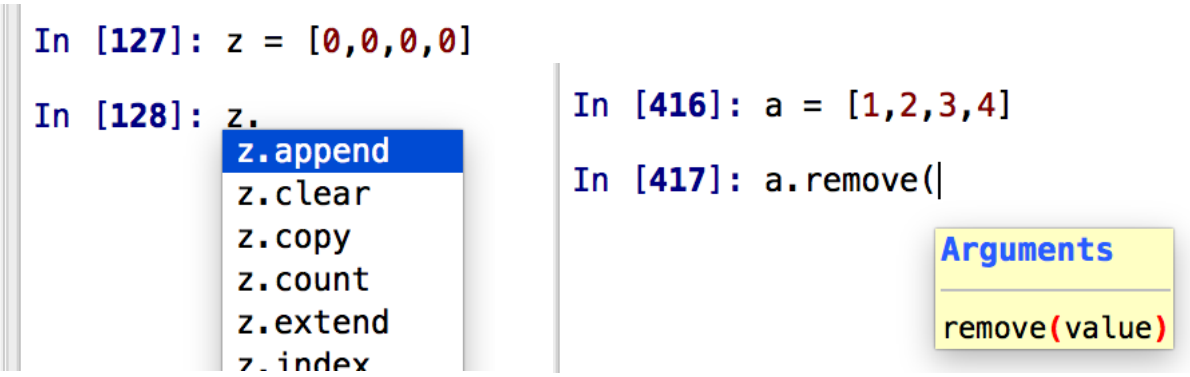
(b) I need a list of running numbers `a = [1,2,3,4,5,6,7,8,9,10]`. How would you do it without hard-coding?

Each character in a string is accessed like a list, but …

```
123 my_string = "abcdefg"
124 for i in range(length(my_string)):
125     print( my_string[i])
126 my_string[1] = "z"
```

What happens at Line 126?

Spyder is your friend – the IPython console gives you information



Some list tricks

Use `dir(list)` to find out what methods are available.

1. Add an entry to the end of a list
2. Join two lists together
3. Remove the last entry of the list and return its value
4. Remove the n-th entry of the list and return its value
5. Remove the n-th entry
6. Remove the entry of the list containing a certain value
7. Reverse the order of entries in the list
8. Sort the entries in a list (all numerical entries) in ascending order

On Spyder console, press . followed by a tab to see the methods available

```
#remove the n-th entry
my_list = ['apple','banana','chicken','durian']
del my_list[2]
print my_list

#remove the n-th entry and return its value
my_list = ['apple','banana','chicken','durian']
entry= my_list.pop(2)
print entry
print my_list

#remove the last entry and return its value
my_list = ['apple','banana','chicken','durian']
entry= my_list.pop()
print entry
print my_list

#remove the entry containing a certain value
my_list = ['apple','banana','chicken','durian']
my_list.remove('banana')
print my_list
my_list.remove('app')

#add an entry to the back of the list
my_list.append('koala')
print my_list
```

solution given here is in Python 2. You know how to adjust it.

Lesson 3 – if/elif/else (2)

Code Snippet 1 and 2 do the same thing. True or False? (sorry its in Python 2)

Code Snippet 1

```
93 if(temperature > 0):
94     print "above freezing"
95 elif temperature == 0 :
96     print "freezing point"
97 else:
98     print "below freezing"
```

Code Snippet 2

```
94 if(temperature > 0):
95     print "above freezing"
96 elif temperature == 0 :
97     print "freezing point"
98 print "below freezing"
```

Lesson 3 – If/elif/else (4)

```
if(x > 7):
    print("Apple")
elif(x > 10):
    print("Banana")
else:
    print("Chiku")
```

What value of x will have Banana displayed?

A. 6
B. 8
C. 10
D. 12
E. It will not be possible.

Lesson 3 – if/elif/else (5)

```
if(x > 7):
     print("AA",end="")
     if(x > 10):
          print("BB",end="")
else:
    print("CC",end="")
```

Which of the following is not a possible output of this code?

A. AA
B. BB
C. CC
D. AABB
E. All are possible outputs

Acknowledgements: kwok shun git

Lesson 3 – While Loop

What is printed by the following code? (Output is on one line to save space.)

```
x = 6
while x > 4:
  x = x - 1
  print(x)
```

- A. 6 5
- B. 6 5 4
- C. 5 4
- D. 5 4 3
- E. 6 5 4 3

Lesson 3 – Lists(1)

```
lst = ['abc', 'def', 'ghi']
lst[1] = 'wxyz'
print(len(lst))
```

What is the output of this code?

- A. 3
- B. 9
- C. 10
- D. 4
- E. No output; there is an error in the second line

Lesson 3 – List Slices (2)

```
a = [1,2,3,4,5,6,7,8,9,10,11,12]
print( a[1:7:2] )
```

The output on the screen is `[2, 4, 6, 8]`. True or False?

Lesson 3 – List Slices (3)

```
77 my_list = [10,20,30,40,50,60,70,80]
78 a = my_list[-4:]
79 b = my_list[-4:-1]
80 c = my_list[-1:-4]
81 d = my_list[-1:]
```

Which of the following lists is empty (i.e. length = 0)?

A. a
B. b
C. c
D. d
E. all have length > 0

More on objects

Recall

Thus far, an object refers to any data in memory.
Each object has an **address** in memory. You can use the `id` function.

You have also learn about **custom data types** using the `class` keyword which have more than one variable that you can treat as one entity or object.

You have seen that we can do operations on a list variable using special commands by somehow "attaching a function to it" using a dot operator.

More on objects

An object can have a set of functions (called **methods**) that define its behaviour.

methods - functions that belong to an object. Each type of object (lists, strings, dictionary) have their own set of **methods**

So, an object is a space in memory that contains data in variable(s) and can also have methods attached to it.

Mutable vs immutable

You have also seen that a string can be treated like a list, but for one big difference.

Objects can be **immutable** (cannot be modified) or **mutable** (can be modified).

You have seen that a list is mutable. Immutable datatypes include string and tuple. Hence in the following lines, which returns a TypeError?

```
my_list = [1, 2, 3, 4, 5]
my_tuple = (1, 2, 3, 4, 5)
my_string = 'abcde'
```

```
my_list[3] = 20
my_tuple[3] = 20
my_string[3] = 'k'
```

The python memory story (4)

Since a string is immutable, why would this code work?

What's the difference between the three lines?

Is data copied?

```
my_string.replace('a','z')
my_string = my_string.replace('a','z')
new_string = my_string.replace('a','z')
```

Advanced Concepts (Not Tested, Good to know)

For those who already know python or Java. Not tested

This is meant for your own exploration.

```
def divide(numerator,denominator):
      if(denominator == 0):
            raise ValueError
      else:
            return numerator/denominator

try:
      print(divide(5, 0))
except:
      print("denominator is zero")
```

Problem Solving

Write a program that takes in a three-digit integer from the keyboard and returns its individual digits.

PCDIT Framework

Problem Statement: Input – Output – Process

Test **C**ases - Generate different inputs and work out the outputs.

Design Algorithm – write down the steps in English

Implement

Testing – test your code with the test cases that you wrote down.

Additional Examples

Taxi Fares

Basic fare	Normal	Limousine	Chrysler
Flag-Down (inclusive of 1st km or less)	$3.00-$3.40	$3.90	$5.00
Every 400m thereafter or less up to 10km	$0.22	$0.22	$0.33
Every 350 metres thereafter or less after 10 km	$0.22	$0.22	$0.33
Every 45 secs of waiting or less	$0.22	$0.22	$0.33

Suppose you take a normal taxi and the **flag-down fare is $3.00**. Write a function that takes in the distance travelled in km and returns the fare payable.

Summing 1 + 2 + 3 + 4 ....

Version 1. How many terms do you need to reach a sum that just exceeds 10000?

What is the sum of 100 terms?

Version 2. Use a loop.

Version 3. Use just one line of code.

Estimating ln(2)

The value of ln(2) can be estimated as follows.

$$\frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} \cdots = \ln(2)$$

Version 1. Write a program that uses this method of estimation to n terms in the series.

Version 2. Write a program that uses this method of estimation to give ln(2) correct to 3 decimal places.

Version 1a (advanced learners). Use maximum two lines of code to estimate ln(2) to n terms.