ENGG1811 Computing for Engineers

Week 9B:
The main() function

Motivation for using main()

- Let us look at the code in my_comp_0.py

```
 9 # Define a function my_comp
10 def my_comp(a,b,c):
11     x = a + b * c
12     return x
13
14 # Test code for my_comp
15 a, b, c = 5, 6, 7
16 output_expected = 47
17
18 output = my_comp(a,b,c)
19
20 if output == output_expected:
21     print('Test passed (v0)')
22 else:
23     print('Test failed (v0)')
```

- It has a function and then followed by some test code
- The first line to be executed is Line 15
- Is it possible to make the initial line of execution a bit more obvious?

Using main()

```
 9# Define a function my_comp
10def my_comp(a,b,c):
11    x = a + b * c
12    return x
13
14def main():
15    # Test code for my_comp
16    a, b, c = 5, 6, 7
17    output_expected = 47
18
19    output = my_comp(a,b,c)
20
21    if output == output_expected:
22        print('Test passed (v1)')
23    else:
24        print('Test failed (v1)')
25
26main()
```

- In some programming language (e.g. C and Java), the computer looks for a function called main() and start executing from the first line in main()
- Although Python does not impose the use of main(), many programmers choose to do that
- An example is in my_comp_1.py
- Note you can replace main by other names

More on importing

- We will use import_attempt_1.py and my_comp_1.my to illustrate a problem
  - This problem also occurs when you use import_attempt_0.py and my_comp_0.my
- We will show you how to fix this problem in import_attempt_2.py and my_comp_2.my

Summary

- Using main()
- Using if __name__ == '__main__’: