

ENGG1811 Computing for Engineers

Week 9C: The main() function

Motivation

- If you look at some Python code, you may see these lines:

```
if __name__ == '__main__':  
    main()
```

- This lecture segment aims to explain what these lines do. We will start with main()

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 (From my_comp_0.py)')
22  else:
23      print('Test failed (From my_comp_0.py)')
```

- 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()

- In some programming languages (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
- The file my_comp_1.py uses a function called main().

```
9   # Define a function my_comp
10  def my_comp(a,b,c):
11      x = a + b * c
12      return x
13
14  def 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 (From my_comp_1.py)')
23      else:
24          print('Test failed (From my_comp_1.py)')
25
26  main()
```

An unexpected behaviour importing

- We will use `import_attempt_1.py` and `my_comp_1.py` to illustrate a problem
 - This problem also occurs when you use `import_attempt_0.py` and `my_comp_0.py`
- Output when `import_attempt_1.py` is run:



```
Test passed (From my_comp_1.py)
Test passed (From import_attempt_1.py)
```

- This printed output comes from `main()` within `my_comp_1.py`
- The file `import_attempt_1.py` does **not** ask `main()` in `my_comp_1.py` to be run but `main()` is executed!

Explanation of the behaviour

- When Python executes a function in an imported file, it will also execute any lines of code in the imported file which are not in any function
- The contents of my_comp_1.py

```
9 # Define a function my_comp
10 def my_comp(a,b,c):
11     x = a + b * c
12     return x
13
14 def 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 (From my_comp_1.py)')
23     else:
24         print('Test failed (From my_comp_1.py)')
25
26 main()
```

The line main() is not within any function, so Python executes it when my_comp_1.py is imported

Fixing the problem

- The contents of my_comp_2.py:

```
def my_comp(a,b,c):
    x = a + b * c
    return x

# Define a function my_comp
def main():
    # Test code for my_comp
    a, b, c = 5, 6, 7
    output_expected = 47

    output = my_comp(a,b,c)

    if output == output_expected:
        print('Test passed (From my_comp_2.py)')
    else:
        print('Test failed (From my_comp_2.py)')

# The following lines mean:
# Execute main() if the file is executed on its own
# otherwise, do not execute main()
if __name__ == '__main__':
    main()
```

We will run my_comp_2.py in 2 different ways: On its own or when it is imported

Changes relative to my_comp_1

Summary

- Using `main()`
- Using `if __name__ == '__main__':` to differentiate between two different situations:
 - A file being executed on its own
 - A file being imported