

Learning outcomes

At the end of this lecture, students should be able to:

- understand that the body of a loop can contain any types of statements including another loop
- show the output of code containing nested loops
- code trace functions which have mutable objects as parameters

COMPSCI 101

Principles of Programming

Lecture 23 – Nested loops, passing mutable objects as parameters

L22 Recap

The user can select a film from a list of titles, and see the film's ratings as well as the average of all the non-zero ratings for that film.

```
{'Jaws': [2, 2, 7, 0, 3, 9, 0], 'The Goonies': [0, 8, 2, 2, 2, 2, 9], ...}
```

```
def process_film_ratings_request(film_list, film_ratings_dict):

def display_numbered_list(list_of_items):
    ???

def get_average_rating(list_of_numbers):
    #see previous code

def main():
    film_list = ["Jaws", "The Goonies", "Aliens", "Commando"]
    number_of_films = len(film_list)
    filename = "Ratings.txt"
    lines_of_text = get_lines_from_file(filename)
    people_ratings_dict = get_people_ratings_dict(lines_of_text)
    film_ratings_dict = get_film_ratings_dict(film_list, people_ratings_dict)
    print("Process Movie-Rating Request")
    process_film_ratings_request(film_list, film_ratings_dict)
```

Nested loops – Example 1

The body of a `for ... in` loop can include any code structures (`ifs`, `if ... else`, `if ... elif`, assignment statements) and they can include other `for ... in` loops or `while` loops. These are called **nested loops**.

```
1 for num1 in range(5):
2     print("A")
3     for num2 in range(3):
4         print("B")
5     print("C")
6 print("D")
```

In total,

how many times is "A" printed

how many times is "B" printed

how many times is "C" printed

how many times is "D" printed

Nested loops – Example 2

How many times is the word "hello" printed?

```

1 def main():
2     for i in range(3):
3         for j in range(4):
4             print("hello")
5 main()

```

Nested loops – Example 3

How many lines of output are printed?

```

1 def main():
2     for i in range(3):
3         for j in range(4):
4             print("hello", end = " ")
5             print()
6 main()

```

Nested loops – Exercise

Give the output.

```

1 def main():
2     number = 0
3     for i in range(3):
4         number = number + 1
5         for j in range(4):
6             print(number, end = " ")
7     print()
8 main()

```

Nested loops – Exercise

Give the output.

```

1 def main():
2     number = 0
3     for num1 in range(3):
4         print(number, end = " ")
5         for num2 in range(4):
6             number = number + 1
7         print()
8         print(number)
10 main()

```

Nested loops – Exercise

Give the output.

```

1 def main():
2     for i in range(2, 4):
3         for j in range(3):
4             print(i + j, end=" ")
5         print()
6 main()

```

Nested loops – Exercise

Complete the output.

```

1 def main():
2     list1 = [5, 4, 3, 2]
3     list2 = [3, 4]
4     list3 = []
5     for num1 in list1:
6         for num2 in list2:
7             list3.append(num1 + num2)
8     print("List3", list3)
10 main()

```

List3:

Nested loops – Exercise

The `get_list_of_vowel_count()` function returns a list of the number of vowels in each word of the parameter list.

```

def get_list_of_vowel_count(word_list):
    vowels = "aeiouAEIOU"

def main():
    name_list = ["Mirabelle", "John", "Kelsey", ...]
    vowel_counts = get_list_of_vowel_count(name_list)
    print(vowel_counts)

```

```
main() [4, 1, 2, 3, 4, 3, 4, 3, 1, 2, 3]
```

Nested loops – Exercise

Give the output.

```

1 def main():
2     for first in range(2, 5):
3         for second in range(1, first):
4             print(first + second, end=" ")
5         print()
6 main()

```

Nested loops – Exercise

Complete the output.

```

1 def main():
2     total = 0
3     for first in range(1, 5):
4         total = total + first
5
6         for second in range(1, first):
7             total = total + second
8
9     print("Grand total:", total)
10
11 main()

```

Grand total:

Nested loops - print_dotted_names()

The `print_dotted_names()` function prints the list of all the names in the parameter list after changing any of the letters of the name which are in the `letters_to_dot` parameter string to a dot.

```

def main():
    names_list = ["Kelsey", "Isobel", "Alistair", "Emmie",
                 "Ophelia"]
    letters_to_dot = 'aeoutsAEOUTS'
    print(names_list)
    print_dotted_list(names_list, letters_to_dot)

def print_dotted_list(names_list, letters_to_dot):

main()

```

```

['Kelsey', 'Isobel', 'Alistair', 'Emmie', 'Ophelia']
['K.l..y', 'I..b.l', '.li...ir', '.mmi.', '.ph.li.']

```

Passing mutable objects as parameters – Exercise

Complete the output.

```

def main():
    a_list1 = [10, 9]
    a_list2 = [1, 3, 4]
    function_15(a_list1, a_list2)
    print("a_list2:", a_list2)

def function_15(list1, list2):
    list3 = list2
    list3.append(list1[1])
    list2.append(list1[0])
    print(" list3:", list3)

main()

```

```

list3:
a_list2:

```

Passing mutable objects as parameters - Exercise

Complete the output.

```

def main():
    a_list1 = [10, 9]
    a_list2 = [1, 3, 4]
    a_list1 = function_16(a_list1, a_list2)
    print("a_list1:", a_list1)
    print("a_list2:", a_list2)

def function_16(list1, list2):
    list3 = []
    list3.append(list1[1])
    list3.append(list1[0])
    list2 = list3
    list2.append(list3[0])
    print(" list2:", list2)
    return list3

main()

```

```

list2:
a_list1:
a_list2:

```

Passing mutable objects as parameters – Exercise

```
def main():
    a_list1 = [4, 3]
    a_list2 = [1, 3, 4]
    function_17(a_list1, a_list2)

    print("a_list1:", a_list1)
    print("a_list2:", a_list2)

def function_17(list1, list2):
    list3 = list2
    for i in range(2):
        list3.append(list1[i])
        list2.append(list1[i])

    list1 = list3
    print(" list3:", list3)

main()
```

Complete the output.

```
list3:
a_list1:
a_list2:
```

Passing mutable objects as parameters – Exercise

```
def main():
    a_list1 = [4, 3]
    a_list2 = [1, 3, 4, 5, 2]
    a_list2 = function_18(a_list1, a_list2)

    print("a_list1:", a_list1)
    print("a_list2:", a_list2)

def function_18(list1, list2):
    list3 = []
    for element in list2:
        if not element in list1:
            list1.append(element)
        else:
            list3.append(element)
    return list3

main()
```

Complete the output.

```
a_list1:
a_list2:
```

Summary

The body of loops can contain any kind of statements including other loops.

Passing parameters which are mutable objects to functions means that the function code may change the object's data.

Python features used in this lecture

```
def print_dots(dot_list):
    for num1 in dot_list:
        for num in range(num1):
            print(".", end = "")
        print()
```

```
for first in range(2, 5):
    for second in range(1, first):
        print("(", first, ", ", second, ")", sep = " ", end = " ")
    print()
```