

# CS1117 – Introduction to Programming

Dr. Jason Quinlan,  
School of Computer Science and Information Technology

**A TRADITION OF  
INDEPENDENT  
THINKING**



**UCC**

**University College Cork, Ireland**  
Coláiste na hOllscoile Corcaigh

Rick and Morty S4 back on E4  
Wednesday 20<sup>th</sup> November...



# Announcements

## Continuous Assessment 3

This Multiple Choice Quiz covers weeks 6 to 11 inclusive

This is a good chance for you to see if you understand what we have covered in these weeks

Scope and List Comprehensions will be on MCQ-2

As we have not completed a Lab in these concepts, I advise undertaking some coding with them

# Announcements

## Continuous Assessment 3

Available only on Canvas

So you will need a laptop, tablet, etc, to take the quiz.

If you do not have one of these, please let me know by email and I will arrange alternative access for the quiz.

If you do not received an email from me allocating you a space with alternative access, you must be in this room to access the MCQ.

# Announcements



## Continuous Assessment 3

Available only on Canvas

You will need access to Eduroam WiFi  
so make sure you have signed up

IP filtering will be used for access to the quiz

# Announcements

## Continuous Assessment 3

Available only on Canvas

A code will be needed to access the quiz

This will be given out at the beginning of the class

# Announcements



## Continuous Assessment 3

Mobile phones will be turned off and placed on the desk in front of you.

You should not access online website for answers during the quiz

If you are seen surfing these sites, you will get a zero grade for this quiz.



# Announcements

## Continuous Assessment 3

This work must be your own,  
so no asking your neighbour for answers

Do not take answers from others machines

There is no guarantee they are correct :)

You can bring pen and paper for rough work

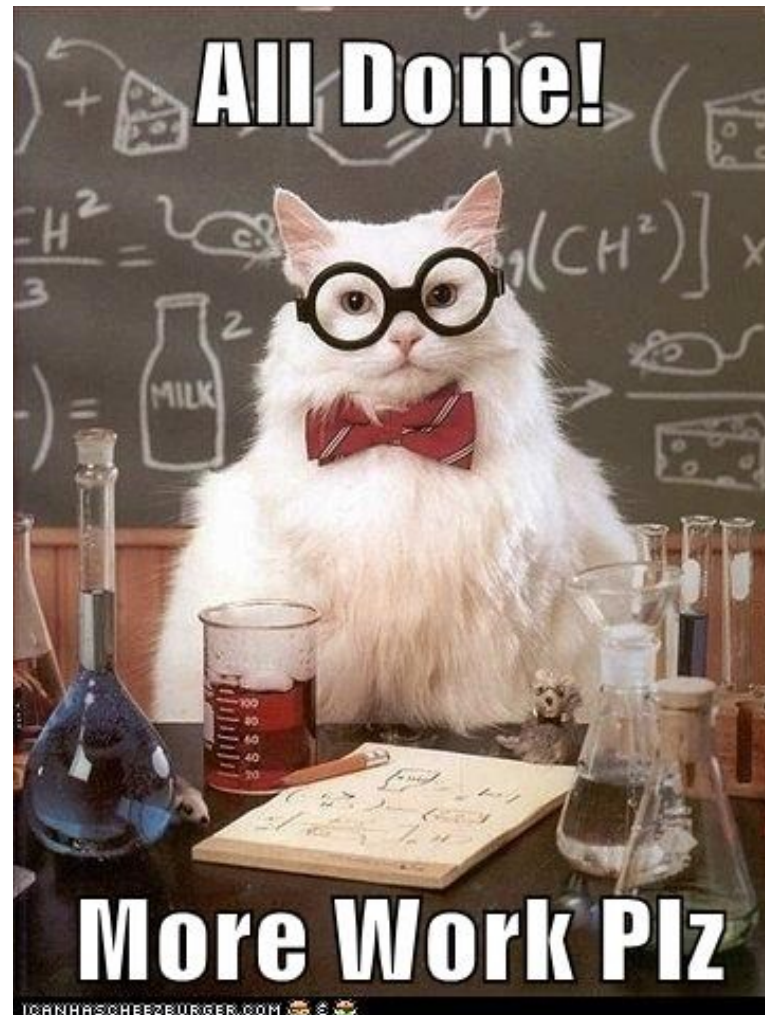
Rough work does not need to be handed up.



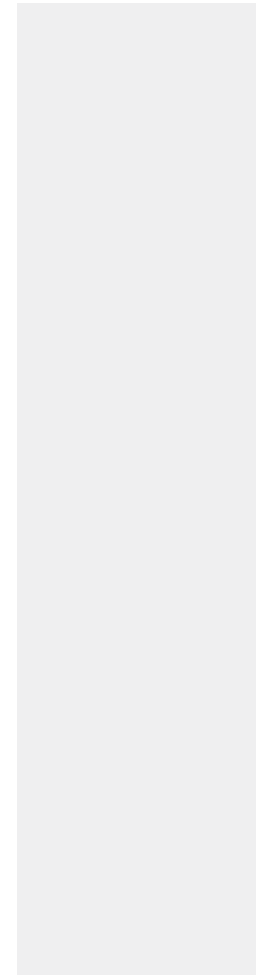
# Canvas Student App

Let's Sign into this lecture now

Access Code  
45734



# Continuous Assessment Lab



# Continuous Assessment Lab

The CA lab was released last Monday morning

Deadline for submission is this Saturday 23<sup>rd</sup> November @ 1am

The lab will be worth 15 marks (5%) of your total marks

This lab is part of 6 labs with 5 of your maximum scores being used for your CA mark

The lab will cover CS1117 weeks 1 to 9 (inclusive)

I will be using “turnitin” – a plagiarism program in Canvas – to check for repeating code between submissions

# Continuous Assessment Lab

The Tuesday/Wednesday labs will be open

But the demonstrators and I will not be able to answer any  
CA coding questions you have

We can only clarify the questions I ask in the CA assignment

i.e., “What do I mean by question X, what is expected as a  
returned value, etc...”

But we can answer any question you have from Labs 1 to 8

# Continuous Assessment Lab



In the CA, similar to the other labs

I will tell you what the functions are called, what the parameters are, and what is expected to be returned.

So make sure you name the functions **exactly**, add the parameters **exactly** and return **exactly** what is asked for...

I will give you six examples for each function call  
and associated expected output

But expect me to test with a lot more function calls, so make sure you test with as many calls as you can think of.

# Continuous Assessment Lab

For the grading, I will be:

1. Calling each of the functions 10 additional times and allocating half a mark for each correct return value. I will not be grading the examples I give you.
2. My tests assume a **returned** value from your functions, so do not use **print**. Print will get you no marks!!!!
3. I will also review each line of your code and if you use Python library functions that you were told not to use, I will deduct all marks for that function. **Only use functions we have covered in class...**
4. If you do not follow all steps in the assignment, I will deduct marks.

# Continuous Assessment Lab

Very, very important:

Make sure you submit on time

Double check to make sure you have submitted

I will not be accepting any of the CA submissions via email



# Continuous Assessment Lab

Very, very important:

Make sure you test your code...

Test you code with my 6 examples

Then think of as many other ways to test it, and retest

Test your code from `main.py` and not `functions.py`

When you upload your `functions.py` file, re-download it  
and retest it.

# Continuous Assessment Lab

To clarify:

```
print(to_english(142))  
# "One hundred and forty two" - wrong  
# One hundred and forty two - correct
```

I do not want *to\_English(n)* to return with quotes  
No need to add `\ "<text>"` to your return string

I state in the exercise sheet:

`return` the string "One hundred and forty two"  
(excluding the quotes)

# Continuous Assessment Lab

To clarify:

When you must **return** multiple values

It is easier to create variables

```
val_1 = <result_1>  
val_2 = <result_2>  
return val_1, val_2
```

Do not **print**

# Continuous Assessment Lab



Best of luck 😊

# Announcements

I have a few concepts left to cover for this semester

Scope, list comprehension, and one or two more

These we will cover this week and most of next week

These concepts will be part of MCQ-2

Live Coding Time...

Scope

# Scope



for loop producing a list from 0 to 9

```
# a list from 0 to 9
x = []
for i in range(10):
    x.append(i)
print(x)
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```



# Scope



for loop producing a list from 0 to 9

```
# a list from 0 to 9
x = []
for i in range(10):
    x.append(i)
print(x)
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print("the next value of i is", i)
```

# Scope



for loop producing a list from 0 to 9

```
# a list from 0 to 9
x = []
for i in range(10):
    x.append(i)
print(x)
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print("the next value of i is", i)
# the next value of i is 9
```

# Scope



while loop producing a list from 0 to 9

```
# a list from 0 to 9
x = []
i = 0
while i < 10:
    x.append(i)
    i += 1
print(x)
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

# Scope



while loop producing a list from 0 to 9

```
# a list from 0 to 9
x = []
i = 0
while i < 10:
    x.append(i)
    i += 1
print(x)
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print("the next value of i is", i)
```

# Scope

while loop producing a list from 0 to 9

```
# a list from 0 to 9
x = []
i = 0
while i < 10:
    x.append(i)
    i += 1
print(x)
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print("the next value of i is", i)
# the next value of i is 10
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x
```

```
print(return_list(10))  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```



# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x
```

```
print(return_list(10))  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is",
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x
```

```
print(return_list(10))  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i)  
----
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x
```

```
print(return_list(10))  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i)  
-----
```

```
NameError: name 'i' is not defined
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x
```

```
i = 0  
print(return_list(10))  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i)
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x
```

```
i = 0  
print(return_list(10))  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i)  
# the next value of i is 0
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

```
i = 0  
x, i = return_list(10)  
print(x)  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i)
```



# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

```
i = 0  
x, i = return_list(10)  
print(x)  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i)  
# the next value of i is 9
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

```
# i = 0  
x, i = return_list(10)  
print(x)  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i)  
# the next value of i is 9
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

```
# i = 0  
my_list, counter = return_list(10)  
print(x)  
#  
print("the next value of i is", i)  
#
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

```
x_2, i_2 = return_list(10)  
print(x_2)  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i_2)  
# the next value of i is 9
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

```
x = [1, 2, 3, 4]  
x_2, i_2 = return_list(10)  
print(x_2)  
# ???  
print("the next value of i is", i_2)  
# the next value of i is 9
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

```
x = [1, 2, 3, 4]  
x_2, i_2 = return_list(10)  
print(x_2)  
# [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i_2)  
# the next value of i is 9
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    # x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    # x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

No errors  
for 'x'

```
x = [1, 2, 3, 4]  
x_2, i_2 = return_list(10)  
print(x_2)  
# ???  
print("the next value of i is", i_2)  
# the next value of i is 9
```



# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    # x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

```
x = [1, 2, 3, 4]  
x_2, i_2 = return_list(10)  
print(x_2)  
# [1, 2, 3, 4, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
print("the next value of i is", i_2)  
# the next value of i is 9
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    # x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

```
# x = [1, 2, 3, 4]  
x_2, i_2 = return_list(10)  
print(x_2)  
#  
print("the next value of i is", i_2)  
#
```

# Scope



Function producing a list from 0 to 9

```
def return_list(number):  
    # return a list from 0 to 9  
    # x = []  
    for i in range(number):  
        x.append(i)  
    return x, i
```

Errors for  
'x'

```
# x = [1, 2, 3, 4]  
x_2, i_2 = return_list(10)  
print(x_2)  
#  
print("the next value of i is", i_2)  
#
```

# Scope



Function producing a list from 0 to 9

```
x = [1, 2, 3, 4]

def return_list(number):
    # return a list from 0 to 9
    # x = []
    for i in range(number):
        x.append(i)
    return x, i

# x = [1, 2, 3, 4]
x_2, i_2 = return_list(10)
print(x_2)
# [1, 2, 3, 4, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print("the next value of i is", i_2)
# the next value of i is 9
```

# Scope



Function producing a list from 0 to 9

```
x = [1, 2, 3, 4]
number = 5

def return_list():
    # return a list from 0 to 9
    # x = []
    for i in range(number):
        x.append(i)
    return x, i

# x = [1, 2, 3, 4]
x_2, i_2 = return_list()
print(x_2)
# ???
print("the next value of i is", i_2)
# ???|
```

# Scope



Function producing a list from 0 to 9

```
x = [1, 2, 3, 4]
number = 5

def return_list():
    # return a list from 0 to 9
    # x = []
    for i in range(number):
        x.append(i)
    return x, i

# x = [1, 2, 3, 4]
x_2, i_2 = return_list()
print(x_2)
# [1, 2, 3, 4, 0, 1, 2, 3, 4]
print("the next value of i is", i_2)
# the next value of i is 4
```

# Scope



So, variable values can be seen in a function when not defined in a function

But not normally when defined in a function when called outside of a function

Scope is a good concept to help understand how variables can impact on your code

Always make sure you pass the correct values as parameters, and return variable as needed...



**UCC**

University College Cork, Ireland  
Coláiste na hOllscoile Corcaigh