Week 1 – Session 1: Programming challenges

We will begin the laboratory sessions for the module with some very simple C++ programming challenges. You **MUST** undertake the directed activities in advance of the class. So program along with the instructional videos and read the directed reading/slides.

Challenge 1: ASCII

ASCII stands for the American Standard Code for Information Interchange and is a character encoding standard for electronic communication. In short it allows us to encode common characters such as '@', '[' and the alphabet as a series of numbers from 0 to 127.

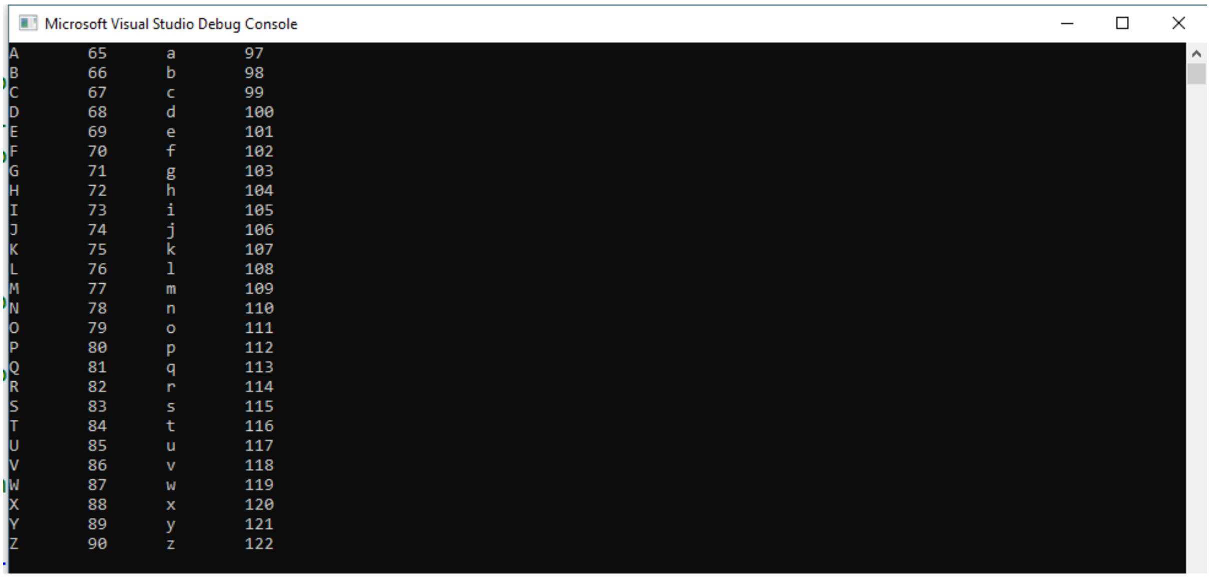
https://en.wikipedia.org/wiki/ASCII

Task 1 Print the ASCII alphabet – duration 10 minutes

The character 'b' is char('a'+1), 'c' is char('a'+2), et cetera. Use a loop to write out a table of characters from 'a' to 'z' with their corresponding integer values.

Task 2 Capital characters also – duration 20 minutes

Modify your previous implementation to also print out the capital letters 'A' through to 'Z' and their corresponding integer values, as shown below.



```
A       65      a       97
B       66      b       98
C       67      c       99
D       68      d       100
E       69      e       101
F       70      f       102
G       71      g       103
H       72      h       104
I       73      i       105
J       74      j       106
K       75      k       107
L       76      l       108
M       77      m       109
N       78      n       110
O       79      o       111
P       80      p       112
Q       81      q       113
R       82      r       114
S       83      s       115
T       84      t       116
U       85      u       117
V       86      v       118
W       87      w       119
X       88      x       120
Y       89      y       121
Z       90      z       122
```

Figure 2: Table of characters and their corresponding ASCII numeric value

Challenge 2: FizzBuzz

In this challenge, ask the user for an integer number n greater than 1. Evaluate each value i between 1 and n.

- If i is a multiple of 3 (but not 5), pint Fizz
- If i is a multiple of 5 (but not 3), pint Buzz
- If i is a multiple of both 3 and 5, print FizzBuzz
- For all others, print the value of i

Task 1 Pseudo code your solution – duration 10 minutes

With a classmate, sketch out your solution using a pen and paper or Microsoft Word.

Hint: You will need a for loop. The modulo operator '%' can be used to get the remainder of a division.

Task 2 Implement your solution – duration 20 minutes

Based on our discussion from task 1 you should attempt to implement a solution.

Challenge 3: Cupcake calculator

Hannah loves to go to CupCakeChaos to by cupcakes (of course!). They are running a promotion that gives a customer a free cupcake after they collect enough wrappers from previous purchases. Determine the maximum number of cupcakes Hannah can enjoy.

Hannah starts with a budget of n pounds. Each cupcake costs c pounds. The promotion provides 1 free cupcake for every m wrappers collected.

For example, Hannah starts with n=4 pounds and buys 4 cupcakes at c=1 pound each. The number of wrappers needed for a free cupcake is m=2. At this point, she can redeem the wrappers and get 2 free cupcakes. She redeems those two wrappers and gets another free cupcake. At this point she has only one wrapper left but she has enjoyed 7 cupcakes.

Write a function called MaximumCupCakes that return an integer, the maximum cupcakes Hannah can eat. Your main function should print this value out

The parameter should take the following parameters:

- n: an integer, Hannah's starting budget

- c: an integer, the cost of a cupcake
- m: an integer, the number of wrappers required for a free cupcake

Task 1 Pseudo code your solution – duration 15 minutes

With a classmate, sketch out your solution using a pen and paper or Microsoft Word. This is going to take a little bit of head scratching

Hint: Break down the task, given n, c, and m how would you determine how many cupcakes Hannah can eat and how many wrappers she has left over. You may want to consider using a recursive function.

Task 2 Implement your solution – duration 20 minutes

Based on our discussion from task 1 you should attempt to implement a solution.