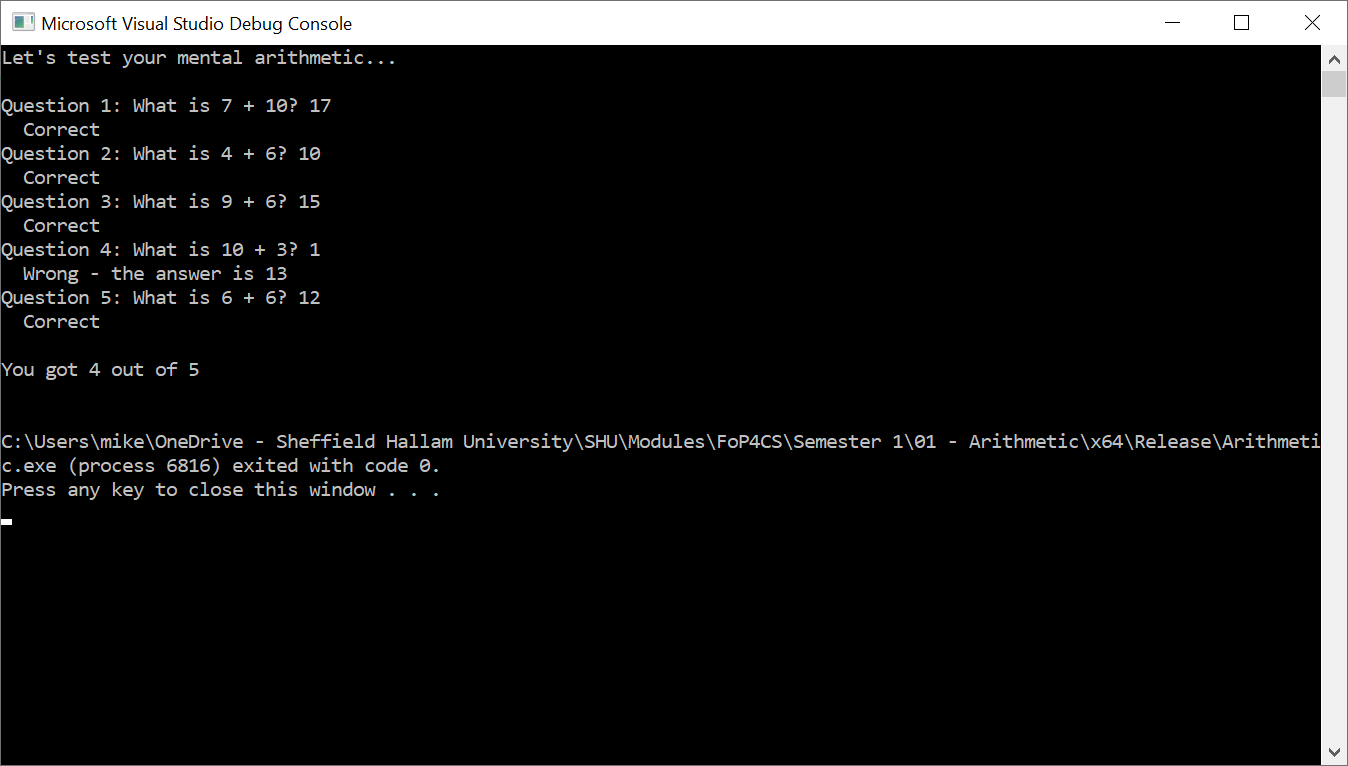
# Let’s Code Again

In this tutorial, we will remind ourselves how to code C/C++ in Visual Studio with some simple applications.

Create the following applications using the good practice techniques learnt from last year. We did something similar in class last year but see if you can code these up from scratch / basics as a coding refresher.

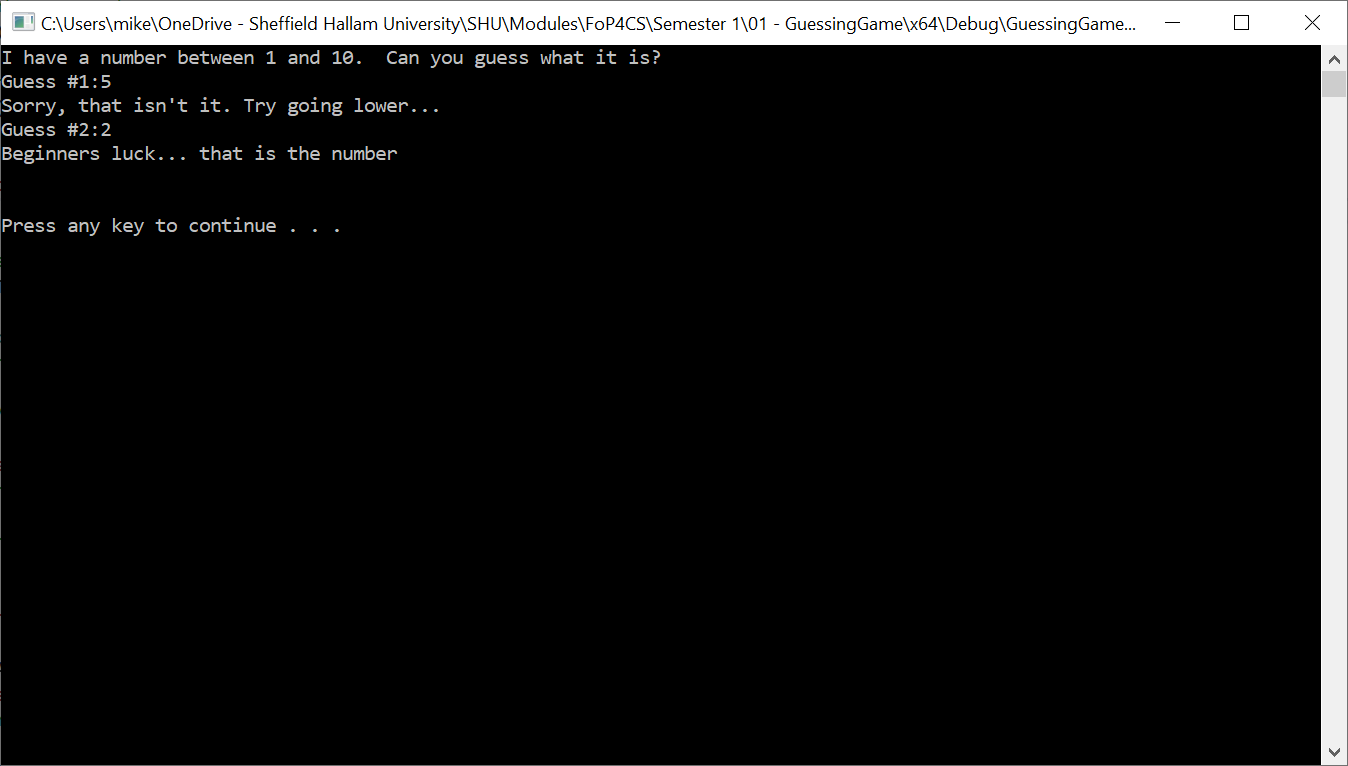
## Arithmetic

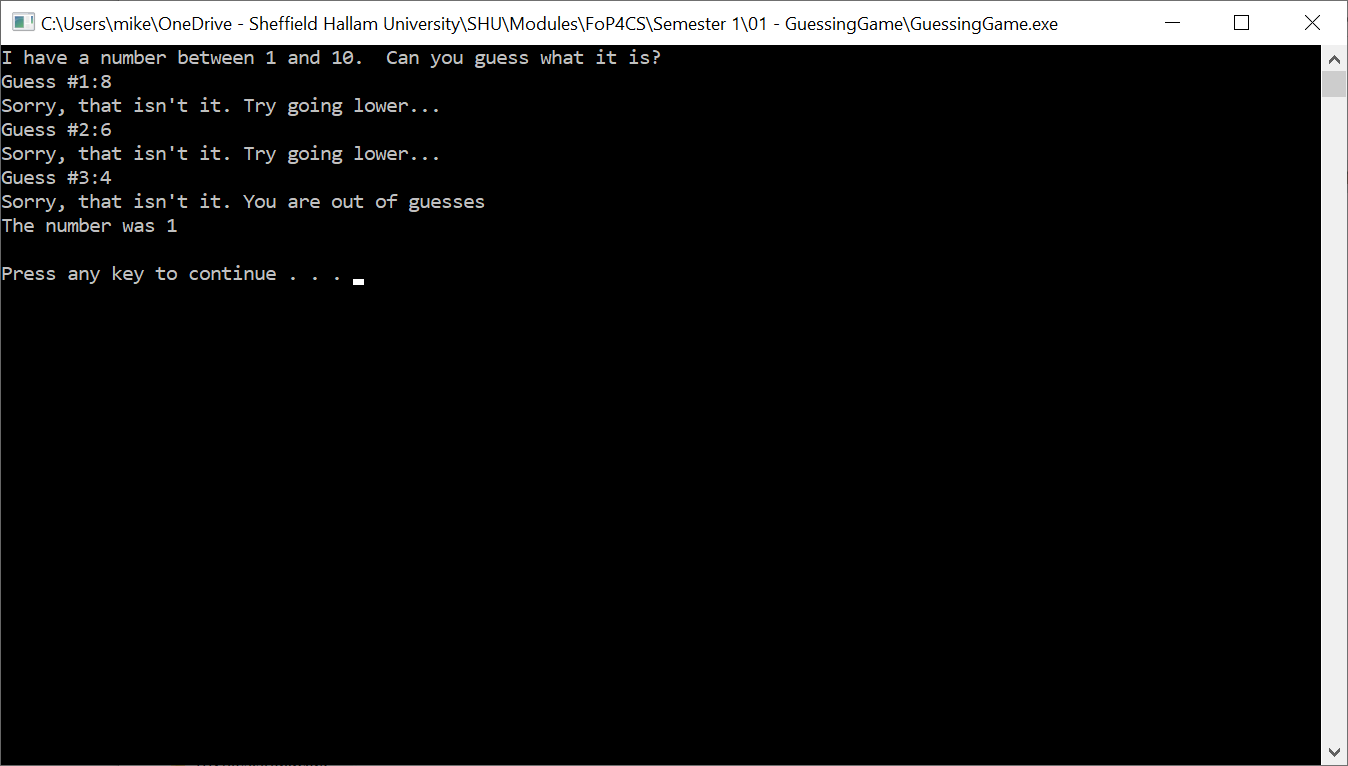
Generates two random numbers between 1 and 10 and asks the user what their sum is. If the user enters the correct answer, they are told they are correct before moving onto the next question. If the user gets the answer wrong, a message is shown informing them of this and the correct answer is given. At the end, a message is displayed that tells the user how many they got right.

Once you have the basic application working, update the code to randomly ask either an addition or subtract question per question.

## Guessing Game

The computer will generate a random number between 1 and 10 and the user has three attempts to guess the number. The user gets feedback at the end of each question depending on the value entered and the number to guess. When correct, a corresponding message is shown and the game ends. If the guess is less than the actual number, the user is told to go higher and if the guess is greater than the actual number, they are told to go lower.





Use the debugger to test your code by determining what number is randomly generated and provide suitable inputs.

## Mastermind

The computer randomly generates 4 digits, each between 1 and 8 inclusive and keeps this initially hidden from the user. The user gets 10 attempts to determine the 4-digit code. On each attempt, the user enters a valid digit combination as a single number input.

Your program should reject any input that contains non-valid digits entered by the user (i.e. 0 and 9) or less / more than 4 digits, asking for another input. You can assume that input will be integer-based so don’t worry about dealing with the user entered in non-numeric characters (we’ll assume the user has some common-sense – usually a big mistake, but for now, meh).

If the user gets the code right a “well done” message is displayed and the program gracefully finishes.

If the user fails to get the code correct, they are told:

* How many digits they have correctly guessed and are aligned with their hidden position
* How many digits they have correctly guess but are misaligned when compared to their hidden position

If the user has failed to correctly guess the code after 10 attempts, they are shown the hidden code and given a suitable commiseration message.

See <https://en.wikipedia.org/wiki/Mastermind_(board_game)> for more information about rules, although we are using digits as opposed to colours.

If you have the game logic for this from last year, try putting it into EasyGraphics and create a graphical version of the game.