**Tutorial 4 – While loops**

**Exercise 1**

* During the lecture you will be asked to write the flowcharts for the exercises 2 and 3.

**Exercise 2 - Guess the number program**

* The solution for exercise 2 should use a while that will loop until the user guesses a ‘hidden’ number.
* **Part A** - Translate your exercise 2 flowchart into a Python program.
* Initialise a variable hidden with a value of 6.
* Ask the user to guess the hidden number (an integer between 1 and 20). Remember to use the int function to convert string input to an integer number.
* If the guess does not match the hidden number the program will run the while loop ‘block’.
  + To state the guess was not correct, and ask the user for another guess.
* Loop while the guess entered by the user is not the hidden number.
* When the guess is correct, exit the loop and let the user know they were correct.

* **Part B** -Amend the program so that the hidden number (between 1 and 20) is a random number generated by the program (check last week’s tutorial for how to generate a random number).
* **Part C** -Amend the program to let the user know if each guess is correct, too low or too high. Use an if, elif, else.

**Exercise 3 -** **Guess the number program (limit number of guesses)**

Translate your exercise 3 flowchart into a Python program.

* The solution for exercise 3 should only allow a maximum of 5 guesses. If a correct guess is entered then use a break to exit the while loop.
  + Write a program that picks a random number between 1 and 20 and allows the user a maximum of five attempts to guess the number.
  + For each incorrect guess, the program will let the user know if the correct answer is higher or lower than the user’s guess.
    1. If the user doesn’t guess the answer in five attempts, the loop will end and the program will tell the user what the number was.
    2. If the user guesses the answer within 5 attempts, the program will break from the loop and tell the user they are correct and how many guesses it took.
  + You will need to add a check directly after the loop to decide which message is appropriate. E.g., either a) or b) above.

**Additional Exercises**

**Exercise 4**

* Write a program that will allow a user to enter a number of scores until -9 is entered.
* When -9 is entered, print the average of the scores entered. Use a while loop.
* Ensure that at least one score has been entered before calculating the average (division by zero would produce an error).

**Exercise 5**

Rewrite exercise 4 so that it uses a Boolean variable (set to True or False) to control the loop. The user will still enter -9 to exit.