C++ Week 1 Control

Exercise 1

Create a new C++ console project within Visual Studio.
Within the main function create a statically declared array named **scores**, to store 100 player scores (of type int).
Within a loop assign each a random score between 0 and 100.
Record the number of scores that lie within the following ranges within a second array named **ranges** that contains 4 elements

- Novice 0..40
- Intermediate 41..60
- Advanced 61..80
- Hardcore 81..100

and display the number of values within each range.
Note: As local variables are not automatically initialised with a default value, you will have to explicitly initialise the elements of ranges.

Exercise 2

We will now create some test code for exercise 1. Add the statements required to sum the values in each ranges and if it equals 100, display a pass message otherwise display a fail message.
Add appropriate processor directives so that the compilation of this test code can be toggled on and off.

Exercise 3

Within this exercise you will create a State Machine that enforce the legal movement transitions a character can perform.

- Define the following movement states as a set of enumerate values within your main function **stand, walk, run, crawl.**
- Define a variable named **state** to hold the current movement state and randomly assign one of the four values. Hint: Typecast a random number to the enum type before assigning it to the variable of the enumerate type. (enumType)(int) where int is the random number.
- Output the random state to the screen.
- The legal transitions are
    - From stand he can either walk or crawl
    - From walk he can either stand or run
    - From run he can walk
    - From crawl he can stand

  Define a switch statement that displays the legal transitions from the random selected value.

C++ Week 1 Control

Exercise 4

Modify your solution to the previous exercise, so that it repeats the random selection of movements 10 times and outputs the movements to the screen. The randomly selected movement should however conform to the transition rules. So if the first state is stand your program should randomly select either crawl or walk.