COM4013 – Introduction to Software Development
Week 15 – Searching, Recursion, and Divide+Conquer
*Always select "**Enable Editing**" if prompted by Microsoft Word*

Lab Exercises

> *Always refer to the lecture notes for examples and guidance.*
> *Also look at your previous work, many exercises are similar.*

There are two "stages" marked in the lab sheet. Stage 1 is the *absolute minimum* point you should reach. Hopefully, you can reach it in the lab session. Reaching stage 2 suggests you are on target for a top-class mark for the module. Set your goals sensibly - do not just aim for the minimum or you may struggle to pass the module.

I use the term integer, string and float, and Boolean to infer what the input/output looks like. Unlike many other languages C++ does expect a data type before the variable names.

So if I ask that you declare an integer num1 to value of 1, then do as follows:

```
int num1 = 1;
```

For declaring a string and assigning the value hello

```
string greeting = "hello";
```

For floats we can declare it like this

```
float money = 2.50;
```

For doubles (more accurate floating-point number) we can declare it like this

```
double money = 2.50;
```

For Booleans (bool) we can declare it like this

```
bool isHeavy = true;
```

Setup

- Create a new Visual Studio Console Application project called **Lab 15 Exercises**. Refer to Week 11's lecture/worksheet/video if you have forgotten how to create a new project/file/program or use the shell code.

Manipulate an Array

- Copy or type the following code into a new project:

```cpp
#include <iostream>
#include <string>
using namespace std;

struct SHighScoreTable
{
    int score;
    string name;
};

int main()
{
    SHighScoreTable highScores[4] = {
            { 40, "ace" },
            { 60, "bob" },
            { 30, "cat" },
            { 70, "ish" },
    };
}
```

- Use a *for* loop to output the contents of the array: score followed by name.

- Search through the array for "bob" and output his score.

- Write a function to output the score of a player, given their name. In other words, the function is passed the array and a name. You search through the array until you find the name. You output the score associated with the name.

    o Test this function on ace

- Write a function to increase the score of a player, given their name and the amount to increase their score by. In other words, the function is passed the array, a name and an integer. You search through the array until you find the name. You increase the score associated with the name by the integer value passed to the function.

    o Test this function on ish

Guessing Game (from semester 1 Week 5)
Write a number guessing game program. You can use the word guessing game from the week 3 and 4 lectures as a basis for your code:

The program selects a number - you can just choose a number and type it in the program (e.g., 231), or better you can get a random number using the c++ random header file (library).

The user guesses this number. If they are wrong the program say so *and tells them whether the correct number is higher or lower than their guess.*

The guessing game continues until the user guesses correctly - a winning message is displayed.

You may as well ensure that the morons who are using your programme can't break it by entering an emoji etc. Gracefully handle the errors.

Times Tables (from semester 1 Week 5)
Use a for loop to display the 12 "times-table" up to 12 x 12:

```
12 x 1 = 12
12 x 2 = 24
12 x 3 = 36
...
```

A little tricky: Update your program to show all the "times-tables" from 1 to 12. You will need to nest your loop.

Optional: Update the program by writing a function that will produce the times tables for any number up to the point that the user defines. The function will take two parameters.

Stage 1   Keep Going...!

High Score Function

- Use SHighScore code from above exercise…

- Write a function to output the name of everyone who has a score higher than a given value. In other words, the function is passed the array and an integer value. Search through the array and output the name of everyone whose associated score is greater than the integer value passed to the function.

- Write a function to **swap** the data stored at index 0 of the array, with the data stored at index 2 of the array. The table will look like the following after the swap:

```
30, "cat"
60, "bob"
```

```
    40, "ace"
    70, "ish"
```
- Write a general-purpose function which is passed the array and two integer values. The integers are indices. Swap over the values stored at the two indices.

Palindrome Recursive Function

- A Palindrome is a word, phrase, or sequence that reads the same backwards as forwards, e.g. madam or 1221.
- I want you to create a recursive function called Palindrome that checks that the sequence reads the same backwards as it does forwards.
- Once you've written the function, use it within an if statement to print out the following messages.

```
racecar is a Palindrome
hello is not a Palindrome
```

Math Recursive Power Functions

- Implement a recursive function that calculates the result of raising a number to a given power.
- Write a recursive function to find the greatest common divisor of two integers.
- Implement a recursive function to calculate the sum of the digits of a given positive integer.

Summation Recursive Function

- Write a recursive function that receives a single integer and returns the summation of every value from one up until and including the integer input.

- i.e., for 5: we would return the sum of 5, 4, 3, 2, and 1 which is 15

Stage 2

Bubble Sort
- Write a function which is passed an integer array.

- The function then swaps over a sequence of values:

  - the values held at index 0 with the values held at index 1.

  - the values held at index 1 with the values held at index 2.

  - the values held at index 2 with the values held at index 3.

- In effect the function you have written implements a version of a "bubble sort". This is an extremely inefficient sorting algorithm. I am not going to teach this yet since it is so inefficient, but it's OK to play with as part of learning to manipulate arrays. What your function does is to cause the first element of the array to slowly move up the array until it reaches the end: this is like a bubble floating upwards, hence the name of the algorithm.

- You could easily enough convert your function into a bubble sort. All it needs is, firstly, the additional specification that the two adjacent values are swapped only if the value held at the lower index (or position) in the array is larger than the value held at the higher index of the array. Secondly, you need to repeat the process of comparing and (possibly) swapping each adjacent value a certain number of times: the number of times is the size of the array.

- It is possible to slightly improve the efficiency of the sorting algorithm by decreasing the range of the array elements being swapped. On the first pass you compare and swap all of the elements, on the second pass you compare and swap 1 less than the size of the array, and on the third pass you compare and swap 2 less than the size of the array, and so on…