Hans Alan Whiburn Haugen

Find my Programming-I repository and Compulsory-II branch here: https://github.com/alanhaugen/Programming-I/tree/Compulsory2

Benefits of using recursion in certain scenarios versus an iterative approach

Recursion is fun to write and results in really neat, short code. However, you can have problems such as a stack overflow with recursive functions. Even though they are neat, fun to write and can make thinking about certain problems easier, if they run wild the program will fail to compile as the program stack reaches its limit. Infinite recursive loops results in programs which simply won't compile. This might be a feature rather than a bug, it might avoid certain runtime errors.

Personally, I find iterative loops easier to reason with and find recursion to be black magick. Therefore, I will avoid recursive functions in my programs.

My progression in programming since the start of the semester

It is a delight to do programming again. I have now not done any programming courses for many years. I am happy to finally follow this track again and hope to be successful.

I have learnt more about program scope, personally I have never seen :: be used to access the global scope. I found an interesting article from Microsoft on scope: https://learn.microsoft.com/en-us/cpp/cpp/scope-visual-cpp?view=msvc-170.

I find Microsoft has the best documentation in the industry. Time to have a European contender to pick up the slack?

I feel inspired to challenge myself with some new programming adventures. I enjoyed programming math problems into the computer and would like to further persue how calculation, particularly algebra and calculus, can be done on a computer.

I still want to learn more about memory. I know there is a stack for non-dynamic variables and for function calls, allocated at a fixed-size at program start, and heap for dynamic allocations. I found the book Data Structures for Games Programmers by Ron Penton to have insight into memory https://cdn.preterhuman.net/texts/math/Data_Structure_And_Algorithms/Data%20Structure%20For%20Game%20Programers%20-%20Ron%20Penton.pdf.

In the appendix there is a chapter called *The Memory Layout of a Computer Program* which I will have to refer to again.

Documentation

I have used Doxygen to generate pretty documentation (XML)

A pdf has been made available here: https://github.com/alanhaugen/Programming-I/tree/Compulsory2/latex/refman.pdf.

For html please see https://github.com/alanhaugen/Programming-I/tree/Compulsory2/html/index.html.