Introduction to C++

Practical 1 (Lectures 1-2)

Either use Compiler Explorer (https://gcc.godbolt.org/), CLion (https://www.jetbrains.com/clion/), or your preferred C++ toolchain. Or try a few different options if you want to experiment.

When you are completing these questions, you may prefer:

- one function per question, with the `main()` function only calling the function you are testing (when you are used to writing functions)
- every question in one `main()` function, perhaps commenting out what you are not currently working on.

1. Create a "hello world" program, and check that it compiles and runs. This is the most important question, so please ask if it is not working!
2. Create and assign values to at least one variable of type `int`, `unsigned`, `float`, `double`, `char` and `std::string`.
3. See what happens when you assign the value `-1` to a variable of type `unsigned`.
4. See what happens when you assign the value `1.23` to a variable of type `int`.
5. Write an `if` statement.
6. Write a `while` loop.
7. White a `for` loop.
8. Generate a uniformly-distributed random number between -1 and 1.
9. Create two `double` variables x and y. Set $x = 0.3$ and $y = 0.4$ and calculate $r = \sqrt{x^2 + y^2}$. Write the result r to the console using `std::cout`.
10. Generate N uniform random numbers x_i and y_i between -1 and 1. Count the number of points where $\sqrt{x_i^2 + y_i^2} < 1$, and use this to estimate the value of π.
11. Code up another estimator for π by calculating the sum of the reciprocals of square numbers (The Basel problem) for N terms, which converges to $\pi^2/6$ for large enough N.

$$S = \sum_{n=1}^{n=N} \frac{1}{n^2} \to \frac{\pi^2}{6}$$

12. Finally, code up the Gauss-Legendre algorithm for estimating π, which has quadratic convergence.
13. Write code to implement the backward Euler method to solve the ODE

$$\frac{dy}{dx} = -y \quad y(0) = 1.$$

on the interval [0 1]. Your code should print a file called xy.dat that has two columns: the calculated values of x; and the calculated values of y. Read this file into Matlab or Python and plot the solution.

[*The backward Euler method for this problem results in the difference relation*

$$\frac{y_{n+1} - y_n}{h} = -y_{n+1}.$$

where h *is step size*]

14. Write code that reads the file xy.dat created in the previous exercise, and computes the error at each point (the true solution is $y = e^{-x}$. Print to the screen the maximum error.

15. Write code to calculate the scalar (dot) product of two std::array<double,3> variables

16. Write code to multiply two 3 x 3 matrices $C = AB$. Think about how you would store your matrices. You could use a flat array std::array<double,9>, or you could use nested arrays std::array<std::array<double,3>,3>. Output the result in a nicely formatted way, for example:

```
C =
| 1, 2, 3 |
| 4, 5, 6 |
| 7, 8, 9 |
```