## Lab Week 2 – Operator Overloading.

# Objectives

The objectives for this week are:

* Understanding and implementing operator overloading.

# Tasks

# Task 1

* As you did last week, clone the folder from GitHub Classroom for this week.
* Take the code and use CMake to produce a solution.
* Go over the code to make sure you understand what’s going on.
* There is an additional utility class to help you output to the console. You can use this or utilise the standard way.
* **Object orientated C++: As we are studying OO, again split the code into a suitable structure that observes the OO paradigm.**

**Task 2**

* Using the code as a base overload all of the arithmetic operators in separate functions for:

|  |  |
| --- | --- |
| Operator | Description |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Modulo |

* Included in the code is a basic Vector3d class. Make each of the operators have the ability to work with this as an input.
* Make sure to check for any infinity / Nan instances in your code.

# Task 3

* After you have completed the above, move on to overloading the:
  + Increment (++) and decrement (--) operators. They need to increase / decrease the whole of the vector3d structure, so it improves and clarifies its use.
  + Equality operators == and !=, so that they compare a Vector3d object (there are a number of ways to achieve this).

# Additional Task (operator overloading)

* Demonstrate operator overloading **outside** of a class.
* Allow the operator that you overloaded outside of the class (perhaps addition) to be global (check lecture or ask if you are unsure of this).

## Lab Week 3 – Error Handling.

# Task 4

* Create an additional function that can catch and throw an exception when going out of range accessing an array.
* Make the array hold Vector3d objects.