## Lab Week 2 – Polymorphism.

# Objectives

The objectives for this week are:

* Setting up the Github repository and cloning the required code.
* Defining polymorphic classes.
* Implementing safe downcasting.
* **(Additional material)** The use of destructors in polymorphism.

# Tasks

# Task 1

* Set up a new folder for this module calling it IMAT2906 \_C++\_AdvancedOO.
* Clone the week 1 lab code repo into the sub folder lab code/week 1 Polymorphism.
* Using CMake, create the solution.
* Simple CMake commands will be covered in the first labs and lecture.
* A solution (one of many possible) will be released approx. three weeks after each of the labs.

**Task 2**

* Look over the code in the solution for polymorphism.
* Take some time to understand what is happening and the structure.
* With the code that you have been provided with:
  + Split it reflecting a more defined OO approach.
  + Create a Player class that is derived from the GameObject.
  + The Player should overload the GameObject virtual functions and also supply another function that can be overridden.
  + Have an additional class derive from Player alongside GoodPlayer (you’ll need to reconstruct this).
  + In the GameObject class, show the two ways that can be used to construct the object using overloading. Comment this.
* Your Main.cpp should:
  + Construct both of the players (or have some form of additional initialising class to do this).
  + Output to the console how the additional classes work (both getting and setting content from the functions as necessary).

# Task 3

* Using the code, demonstrate the use of upcasting, focussing on one of the two player classes.
* Comment this, demonstrating how it works.

**Task 4**

* Create destructors for the GameObject and newly created Player classes so that they align with being used as polymorphic classes and would not produce possible memory leaks.
* If you are unsure of the process for this, take another look over the lecture notes and videos.