**A Short Report on Darts 501 Assignment**

Authored by D. Green.

Student No. 1900931



UNIVERSITY

*of*

ABERTAY DUNDEE

# Summary of Approach

Taking what was learned from the week 6 assignment, the approach was to extend what already existed in the 301 bull only form. The pseudocode was adapted as was the UML, and work within Visual Studio began rather quickly. Due to having put a reasonable amount of effort in at that early stage, an MVP already existed.

It was an iterative process overall, although much less than in assignments completed in previous years, again being due to early efforts in the 301 assignment. This will have genuinely been one of the very few occasions that pseudocode was completed first, with time spent trying to get an algorithm to work on paper first before even opening an IDE. The temptation to ‘just get going’ was great but was thankfully resisted. This helped immensely, it took patience and a great deal of problem-solving, however, paid dividends when it came to writing the solution. While the result is not a replica of the pseudocode, it can be said that due to the solid framework and algorithmic understanding that this provided it made writing the solution in VS that much easier. With only limited problem solving required while working in VS more time was able to be spent focusing on other details such as language, API or syntax issues, it is enough to concern oneself with these alone without also having to try and design and problem solve at the same time.

It was very quickly found, certainly as it is understood, that the simple solution was complete and from there, and due to how the software had been designed, it was not too much trouble to insert additional functionality to try and meet the criteria for the enhanced solution. Again the pseudocode was adapted and the UML also, and from there the solution updated, and as it is understood the enhanced solution criteria were met. This process was once again repeated for the interactive solution, and again due to the design, it was a relatively straight forward process of simply prompting the user to enter which number and which type they wish to ‘aim’ for, and then passing this information to the already existing functions and letting the program run as before. It was quite surprising to find new code was able to be integrated rather easily for the enhanced and interactive solutions without breaking things, of course, some small modifications were required to accommodate the new code but these were small and trivial.

# Reflection on the Benefits of OO Principles

As has been touched upon, it seemed that through design choices and object-oriented principles it was made simpler for further enhancements to be made without upsetting too much of the already existing code base due to the nature of its compartmentalization. It is difficult at this stage to make a direct comparison between what has been completed previously and what has been completed now due to now having been using OOP for some time.

However, to provide a perspective of Procedural Programming vs Object-Oriented, as it is understood, could be summed up as follows. Object-Oriented Programs are simply collections of organized and compartmentalized procedural snippets. This of course massively helps with organization and maintainability and for me was evident in how easy it was to add new functionality without either having to change too much of the already existing code or it completely breaking things.