Coding constructs and techniques

# Task descriptions

The program will need to be able to collect team names at the start of the process.

The program will need to be able to collect player names for that team (e.g. Team 1 will have player X in it).

The program will then need to be able to collect score’s **each** event for **each** team, storing them in an array temporarily.

After all events are finished, an overview of each event will be displayed, showcasing the scores for each team per event.

These scores will then be totalled for each team. The program then needs to sort the order of the scores in order, highest to lowest, and then output them.

# Algorithms

The algorithms present in the program do not need to be overly complex; however, some basic sorting and comparison operations, such as sorting the teams on their total scores in descending order using the Array.Sort() method, or comparing two integers using the CompareTo() method will be required to accomplish the descriptions outlined in the task description.

# Data structures

In order to store both team names and players, a two-dimensional array will be required. Furthermore, to store the scores for each event and each team, we will also require a two-dimensional array, where the first dimension represents the events, while the second dimension represents the teams.

A standard array (1 dimensional) will also be utilised to store the total score of each team.

Various integer variables will also be implemented to store the number of events and the number of teams. This will most likely be stored as a constant.

Lastly, several string variables will be implemented to store team names and player names.

# Data storage

The code will not currently make use of any persistent data storage, such as databases, files or network storage. Instead, the data is stored in memory using various arrays and variables; however, once the program finishes execution, the data is lost and needs to be re-entered.