Reflective Report

To finish this course work I used 4 different classes: Mountain, Climber, Club and ClubStats, following the specification. Mountain, Climber and Club are the basic classes. ClubStats has the main method to execute all questions.

Mountain class is the class which stores a mountain's name and height. It has methods to get name and height. And it also has methods to set new name and height. I use 'int' type to store height, because 'int' type is easy to compare and compute. The main issue with using an 'int' is its inaccuracy, therefore an improvement that I could have made to the program is implementing a 'double' or 'float' value. The mountain test class tests the get and set methods.

Climber class is the class which stores a climber's details and has methods to return details of climbed mountains. In all the methods returning a mountain's details, I first check if there is no data stored for the method to return. If there are no climbed mountains, the getAverageHeight() method will return 0, and the other two methods will return 'null'. In getGreaterList(), if there is no mountain higher than the height level the method will return 'null'. The reason I have chosen to use 'null' because of its versatility. The climber test class tests all the cases of the mountain details method, like no mountain case, no greater than height level case.

Club class is the class which has methods to return the Club's recorded details. In all the methods the program also checks if there is no climber. In the getHighestAverageClimber() method, There is a possibility that two climbers could have the same average height and to prevent this I chose to create an ArrayList to store the climbers if the highest average height was shared by more than one climber. I think the getHighestMountain() method needs to return both the mountain and the climber who climbed the mountain because of the Clubstats class requirments, so this method uses a HashMap datatype to store the information. This also takes into consideration that different mountains may have the same height, the HashMap will store them separately. In the getGreaterList() method, there is a possibility that different climbers could have climbed the same mountain so this method removes the repeat data. The club test method tests all the cases of the mountains climbed by different climbers, like no mountain case, no climber case and all relevant boundary conditions.

The Clubstats class contains the main method. For purpose of clarity I have decided to implement 3 methods for the menu. Each method executes a single option, which I believe make the code easier to read and understand. All input is checked by the method prior to being stored to verify validity. For the option two, the program gives the user a list of climbers and allows the user to choose their desired climber. An improvement that could be made here is to give the check type methods a specific package and import them into Clubstats class to improve clarity and readability.

In conclusion, I used many data types like HashMap and ArrayList. Most methods contain lots of conditions to make the program more accurate. However, there is still a room for improvement, like changing data types, file structure and adding algorithms.