**REFLECTION -COURSEWORK 2**

Coursework 2 was perhaps a complex one, i basically approached it in a more object oriented manner to define the various functions that will necessary help in determining the actual components of the coursework. The requirements of the system were comprehensively intensive and as such the actual implementation was a thought process. The calculation of the weighted overall score to rounding off to the nearest category was one of the key step and were fairly complex and intensive. I was able to focus on each functionality and isolated the various steps and parts of the program as needed. The major categories that I isolated were, calculating the overall score, rounding off the the score to nearest category boundary, and lastly determining the corresponding category. Each module plays a crucial role in the functionality of the program. Each function has a clear singular functionality. By splitting the program i am able to test the program, make amendment where necessary and also debug in cases where there are errors. At first i also initialized the various data marks and bounds for the various categories.

The flowchart also helped in determining the various bounds that needed to be used on the decision making for the various marks bounds. It gave a clear road map of the various utilized functionalities that necessary helped in determining the functions and how to actually implement them. I understood the sequences of the program even before starting the implementation since the flowchart already mapped out the functions indirectly.

The major challenge that i encountered during this implementation was the the designing of a reliable rounding mechanisms that would assign a correct score and also map it to the neatest category. I tried mapping this functionality with an if-else statement but the more i tried to fix that the more i got more problems. So i switched to a more data driven functionality where, i can basically use tuples and also represent the various score boundaries correctly. By matching the correct boundary using their absolute differences i possibly could enhance the programs functionality and also adjust on the accuracy and the way i was implementing the tuples and how they work.

The use of a robust input methodology was critical though this implementation. As such, it was comprehensively the hardest part, since i had to ensure that the users entered only numeric values and that this values should be in the range of 0-100. It was critical to also check for spaces and clear them out if the rest of the input was only numeric. I used the try-except functions correctly to map this out.

The rounding\_to\_nearest\_category() functions is the interesting function that i have had to design the whole of this coursework since, i had to review how lambda works and also how its functionality would help in determining the various conditional approaches within the coursework. Given more time, i would concisely enhance the program by allowing multiple student at once and also storing their scores. File handling would also be a much of a good functionality in this implementation to ensure that there is easy management and scalability making the program more real-world oriented.

In summary the implementation of this coursework enabled me to understand more about modular programming and how to implement structured coding. By doing this i have understood the adaptability of the various specified requirements choosing and also how to map them into functions and implement them into correct functions.