Evaluation

For this task, I had to make a program that allowed a user to submit information about their latest plays on a game. This information included a unique player ID, their number of plays, and their play time and score for each game.

Planning

I feel that I have done the best planning possible for this task and its given scenario. I started off by making a flowchart which I managed to get to a high standard after only a few tweaks when I noticed errors. To the best of my knowledge, I have stuck to all the required conventions and standards while making this flowchart. With the aid of this flowchart, I proceeded to do a pseudocode version of the program to better visualise how it should be laid out to better prepare me for the program development. My flowchart shows any calculations, data storing, and variables needed; however, I did change the layout of the code to be cleaner and more understandable.

Development

During the development of the program for this task, I sectioned off each different stage into individual functions so they can be called as needed and I can avoid over-nesting my code to keep it tidy. I have added fitting comments that show what each step of the program is doing in detail. I have been consistent with my variable and function names, deciding to go with camel case as it is the least visually complex option. My code is well organised and flows in a way that makes sense to read. I encountered a few errors while developing the program, but I thought through them logically and was able to implement fixes for all of them (this can be seen in my testing documentation). The task 4 program features data validity checks which serve as a way to make sure that the program will only accept reasonable data that can be used in later calculations. I firmly believe that my code is compact with as minimal repeated code as possible for the task.

Test Plan and Testing

My test plan shows that I have thought of many issues that could show up, some of which turned out to be true. As I tested my program against the test plan, I made sure to work on any errors as I discovered them to help reduce any loss in functionality in my final upload. I have listed any errors and the fixes used in my testing documentation. I do wish that I had created more tests to perform on the program to ensure that there

were absolutely no errors or issues present at all anywhere, but I struggled to think of tests that were different enough from each other, so they weren't pointless or repetitive.

Changes

In my pseudocode, I said that the unique files would be saved to a folder called "scores". While I was making my program, I realised that making folders was far out of my abilities and decided not to proceed with it. I did still include the unique files, but now they just get saved to wherever the program is running. I also showed that there would be a function called "charCheck", but this didn't end up working out as I had to check for many different things which would end up creating more code than what was needed if I just checked individually.

Summary

My code does work for the given scenario, but there are ways that I could make it much more efficient. I could have been a bit more detailed with my comments to provide more context on how the code actually works. While my flowchart is clean and does show a baseline of what I wanted to do for the programming section, I still feel that there were better ways in which I could have planned in the initial step that would have given me a great boost when doing further steps.

Overall, I think I have performed well for this task and scenario. Despite a few setbacks, the program works as expected. The issues that popped up have helped me expand my knowledge of Python and I will be sure to use this in any future projects.