

School of Computing, Creative Technology and Engineering

**Module: Fundamentals of Computer Programming**

**Academic Year: 2023/24**

**Level 4: Semester 1**

**Assignment Title: Project f1-Documentation**

**Date Due: Jan 16, 2025**

**Tutor: Saurav Gautam**

**Student Name: Sange Doma Tamang**

**Student ID: 10260**

GitHub Link: <https://github.com/Sangedoma/Programming-Portfolio/tree/main/project-f1>

**Project f1-Documentation**

**Introduction:**

The project follows the analysing of a file that is generated by the timing system of different lap times at a Formula 1 Grand Prix. It processes Formula 1 lap time data from one file and compares the performance of drivers based on their fastest lap times. The program works with 3 lap time files: laptime1.txt, laptime2.txt and laptime3.txt, which consist of data for three separate laps that took place. For the lap txt files, the files were modified with spaces after the driver’s code names as it avoids any risk of misidentifying a driver name with their team name. There is an additional file called drivers.txt which contains further details for the drivers like their full name and the name of the team they belong to.

**Functions Used:**

**convert\_time\_to\_secods(lap\_time)**

This converts a lap time from a string format into seconds, which makes it easier to compare and calculate the averages for the drivers for each lap. Each lap time in the files is associated with a driver’s code. The lap\_time string is split into minutes and seconds using (:). When the program reads a line from the lap list, it splits the line into two parts, Driver’s code and time taken. Once the lap time is converted into seconds, the program displays fastest lap time and average lap time.

**load\_driver\_details(filename)**

This function loads the driver’s additional details from the drivers.txt file into a dictionary. The filename here, is drivers.txt as it contains the driver information. First of all, it checks if the necessary file exists using os.path.exists(). When it is confirmed that it exists, the file opened and each line gets read. As the file consists a driver’s details in: ID, Code, Name, Team format. The line gets split using comma and the details get extracted. Likewise, driver\_details dictionary gets created where the key is driver’s code and the value is a tuple containing the driver’s name and team.

**process\_lap\_times(lap\_filename, driver\_filename)**

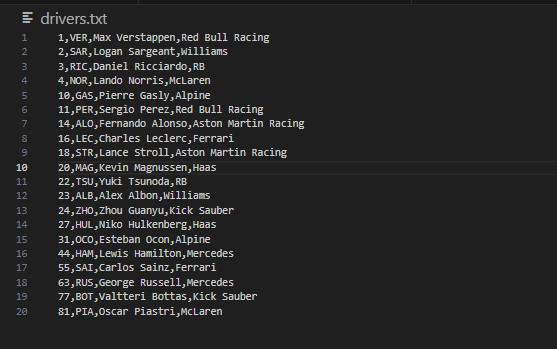
In this, first it checks for the availability if the file, lap\_filename exists or not. If it doesn’t then it prints an error message. Then the function loads the driver’s details using the load\_driver\_details() from the driver\_filename. It reads the lap\_filename and gets the race location which is the first line of the file. The lap times get stored in a dictionary namely, (driver\_times). The fastest lap time and average time for each of the drivers get calculated. In descending order, the drivers with the fastest lap time get sorted and the result gets printed in a table format. The fastest lap time for the corresponding race also gets printed.

**if \_\_name\_\_ == "\_\_main\_\_":**

This part of the code ensures that the process\_lap\_times() func in only called if the script gets run directly as, not imported as a module. It checks if the execution of the script is being executed with the correct number of arguments. When the correct number of arguments is provided, it calls process\_lap\_times() with the filenames.

**Files created:**

**Drivers.txt:**

****

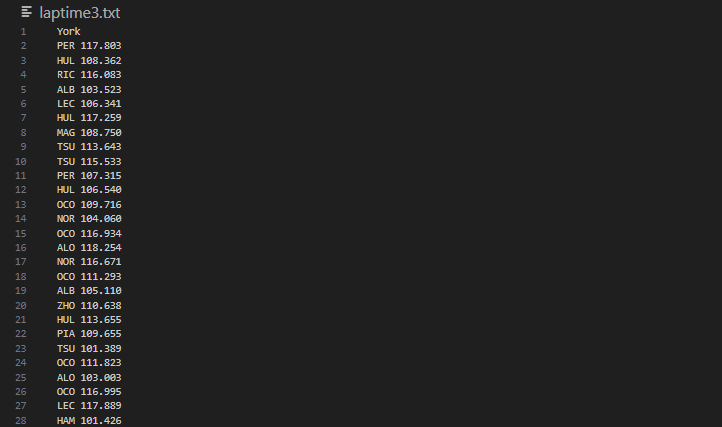
**Laptime1.txt:**

****

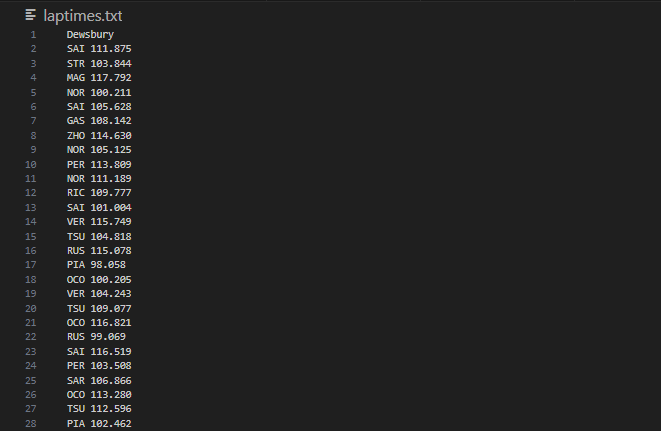
**Laptime2.txt:**

****

**Laptime3.txt:**

****

**Laptimes.txt: (extra created by combining all the 3 laptimes.txt)**

****

**Error handling:**

In this program,

If the lap times file is not found then it will print Error: Lap times file does not exist.

If the lap times file is empty then it will print: Error: Lap times file is empty.

Invalid lines in the lap times file will be skipped with a message: Skipping invalid entry: <line>.

Invalid lap time formats will be skipped with a message: Skipping invalid time format: <lap\_time>.