Assignment 1 Report

I decided to approach this assignment in a unique way by implementing my knowledge and logic understood from highschool. As most would potentially create functions to handle each specific task with further elaborations necessary, I decided to structure my work in a tidy manner. Additionally, I originally had all my work in one Main class instead of another class because a lot of the tasks could be done by scanning the file instead of looping through the data countless times. However, due to the Professor's wish two hours prior to the deadline, I had a brief experience of what it truly means to be an anxious programmer. This essay explores the functionality and significance of my Java program that processes information from a CSV file, creating a collection of GooglePlayStoreApps objects.

The primary objective of the program is to read a CSV file containing data about various Google Play Store applications. The CSV file is structured with each line representing details about a specific app, including information such as app name, ID, category, rating, installs, and more. The goal is to convert this raw data into a more manageable and structured form, utilizing Java's object-oriented capabilities.

At a first glance, the work is straightforward and handles circumstances in a simpler manner. I believe that a programmer must develop this crucial skill to simplify analyzing codes from another's perspective; making the logic of the program structure understandable. Then again, every programmer approaches a problem differently. For example, counting how many free apps there are, was completed by scanning the file instead of looping through everything.

To ensure data integrity and handle potential errors, the program incorporates various error-checking mechanisms. For instance, I decided to think outside the box and read data from the altered CSV file I made of 100,000 lines of code instead of the memory-consuming over 2.3 million lined CSV file. Moreover, I had two try-catch functions to scan the CSV file and check for errors in the data provided; e.g. this may occur if there are issues converting certain string values to numeric types.

The output phase showcases the program's ability to distill meaningful information into CSV files. Files like AppsPerCategory.csv illuminate the distribution of apps across categories, Top100Companies.csv highlights prolific app producers, and TopDevelopersWithoutCompanies.csv identifies top developers irrespective of company affiliations; each task has a CSV file. Additionally, the program calculates the number of apps purchasable within specified budgets ($1000 and $10000) and offers a breakdown of free and not-free apps in FreeNotFreeCounts, one CSV file per budget is offered. This structured approach to outputting results enhances the clarity and utility of the generated data.

Code organization has been a priority throughout development. Clear and concise method names, coupled with comments explaining the purpose of each section, contribute to the overall readability of the code. Leveraging Java's Stream API for sorting and collecting data into maps not only aligns with modern coding practices but also enhances the program's efficiency. The

inclusion of error handling, notably for potential file absence, with a corresponding stack trace, adds an extra layer of reliability to the program. Additionally, the creation of helper methods, such as those for sorting companies and developers, enhances code modularity and ease of maintenance.

In conclusion, this Java program represents my effort to adhere to good programming practices in data manipulation, error handling, and output generation. Its clarity, modularity, and adaptability make it a valuable tool for extracting meaningful insights from Google Play Store app data. Regarding the question of difficulty, I personally only had one challenge in figuring out an appropriate regex for the types of errors in the CSV file. If I were to complete this assignment again, I would make the Google PlayStore Apps Class on time (not two hours before the deadline.) Therefore, it would be helpful to develop some sort of plan to distribute the process of this assignment easily. Nevertheless, there are many opportunities to work on this drawback this semester.