Assignment 1 Review

In order to complete the task, I wrote the main function from which all other functions are called, the additional functions that perform some actions, as well as an App class which helped me finish the fourth task. I had 4 different hashmaps, and I wrote a function called getAllMaps(filePath, categoryAppsMap, companyAppsMap, developersMap, downloadsMap), which went through the “Google Play Store Apps.csv” file. With this one function, I was able to extract all the information needed from the file into all my hashmaps. The reason for why I did this is so that the scanner would only have to pass through the file once, as the file is large and it would take a lot of time for the scanner to go over the whole file for each map individually. The most challenging part about the assignment was to split the lines of the file, so I found a regex that was able to properly separate all lines. The regex I used is ",(?=(?:[^\"]*\"[^\"]*\")*[^\"]*$)" which is a regex that matches a comma, but it uses additional rules: ?= (Positive Lookahead) means that a comma should be followed by something specific, but not to include it in a match, ?: (Non-capturing group) which is used for grouping elements together without making a separate capture group. This means that organizes the pattern inside and ensures that the content inside of this part is not captured independently. This part of the expression (?:[^\"]*\"[^\"]*\")* matches zero or more double quotes which are followed by any character that is not a double quote, and lastly [^\"]*$ matches any number of characters which are not double quotes until the end of the line. The part that I thought would be hard but it was actually easy was the second task, specifically determining which developer works/does not work for a specific company in the third task. The way I resolved this issue is by splitting the email using regex @, and then I used the \\. Regex to split the parts after the @, followed by a function which reverses the order of these parts. This way, I was able to check if the developer works for a company or not using the AppID column and contains() function. After the function getAllMaps, I had 4 maps (3 of them are <String, Integer> and 1 is <String, Long>). I wrote separate functions that would write the necessary information to files, as there were processes such as sorting which needed to be done beforehand only for some of the maps (for example top 100 companies with most apps developed). These function are mostly straightforward, as they use key,value pairs from the maps in order to write specific data to the files. The additional processes that were necessary apart from sorting were the replacements of all characters with empty strings except from digits and dots using the regex [^\d.] and the replace() function, which was done for the fourth task in order to be able to parse the values from the price column of the file. In all functions where I had try and catch, I print the error messages if there are any to the terminal in order to see which lines are skipped, which were some empty strings in my case. If I had a chance to do something differently next time, I would make one function for writing to the files, and if there sorting or some other function is necessary beforehand, I would do it in the main function before calling the function which writes to files.