Owen Gresham

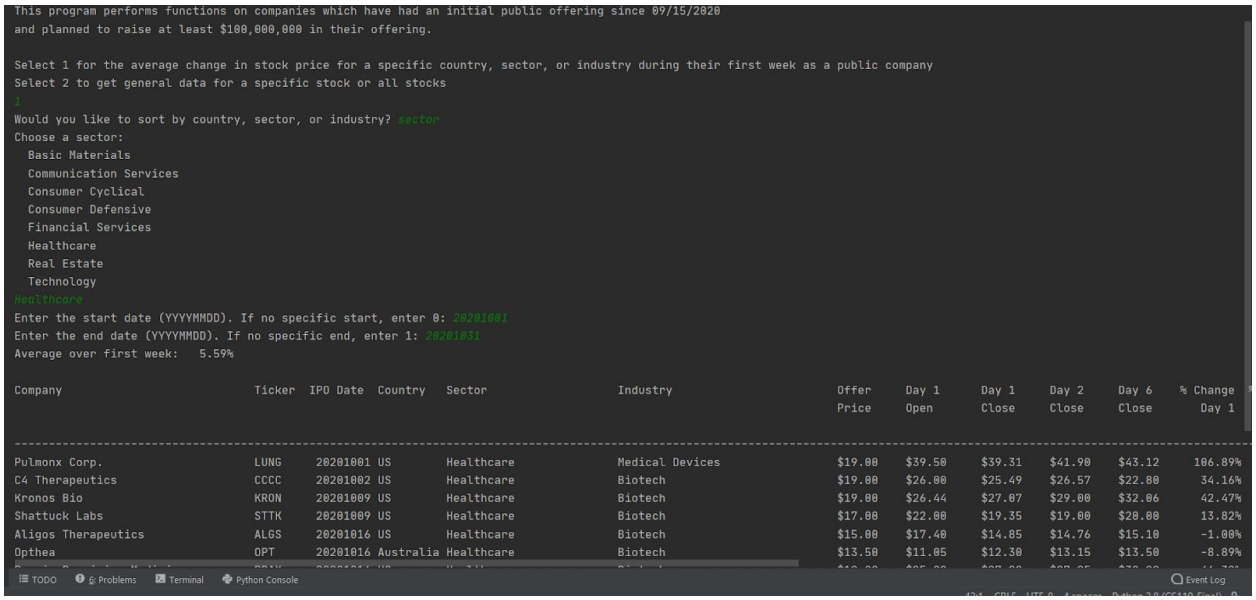
12/13/2020

CS-110-A Final Project Report

Section 1: Overview and Summary

This program is based on a spreadsheet that I have kept up to date for the last few months. The data in the spreadsheet (initially containing many more columns, simplified for the sake of this project and converted from .xlsx to .csv) contains data tracking the performance of a company in the first week after it has completed its initial public offering (IPO). The spreadsheet contains, for each company, its IPO date, sector, industry, home country, prices on different days, the percent change in the stock price on each day for the first six days it is a public company, and the percent change of the company over the entire week. I have covered the price change over the first six days so that I know how the stock price has changed one week after it has gone public. For example, if a company went public on a Monday, the spreadsheet tracks how the stock performed up until the closing price on the following Monday.

The program creates an object for each company, and each column is an attribute of that Company object. The only methods of the Company class are the getter methods for each of its variables. The main() method prompts the user to choose one of two options. The first option allows the user to choose to see the performance of companies over their first week public based on either their home country, sector, or industry; the user can also choose to limit the companies to within a certain timeframe. The second option allows the user to enter a ticker symbol and have neatly displayed the information contained in the spreadsheet. The second option also allows the user to display all companies contained in the spreadsheet over any time interval.



Section Two: Target Audience

The targets of this program are trend following stock traders. The performance of companies after they IPO changes. During some months or years these companies may initially perform great and during some months or years they may initially perform poorly. Traders can use this program to see how they are performing during a specific time period or even see how certain types of companies are performing after going public.

Section Three: Programming Techniques

Object-oriented programming is used as the basis of the program. Much of the time spent writing this program was devoted to the methods used to neatly display the data. One method takes a Company object as a parameter and formats all of its instance variables so that the method can be called for multiple companies and their data can line up neatly in columns. Since there are different menus that appear throughout the program, many if-statements were used. Each option at the start of the program has its own method. Another integral part of the program

is the method that creates a list of Company objects. Since a Company object has many variables, a for-loop iterates through the CSV file and creates an object for each row, assigning the value in each column to its corresponding instance variable.

Section 4: Challenges

Writing the methods that formatted the variables of a Company object was very tedious and time-consuming. Another challenge I encountered during the program was the large amount of if-statements and for-loops that often made it difficult to keep track of the order that the code was being read in. There was a large amount of troubleshooting due to the confusion caused by this.

Section 5: Future Extensions

There is a lot of data in the CSV file, allowing for much improvement to the program. For example, since each Company object already contains the percentage change of the stock on each of the first six days of the company being public, I could ask the user to enter a time interval over those first six days and the program would return the company's performance over those specified days. This could also be used to compare how specific sectors or industries perform on certain time intervals over the first six days. Another addition could see how non-American companies perform compared with American companies since most companies in the spreadsheet are America-based.

```python
def main():
    data = open("C:/Users/oweng/PycharmProjects/CS110_Final/IPOcsvData.csv")
    companies = create_company_list(data)
    print("This program analyzes companies which have had an initial public offering since 09/15/2020")
    print("and planned to raise at least $100,000,000 in their offering.\n")
    print("Enter 1 for the average change in the stock price for companies of a specific country, sector, "
          "or industry during their first week as a public company")
    print("Enter 2 to get general data for a specific stock, or all stocks that had an IPO in a specific time frame")
    selection = input()
    if selection == "1":
        chosen = firstChoice()
        print("\n", performanceBy(companies, chosen[0], chosen[1], 18, int(chosen[2]), int(chosen[3])))
    elif selection == "2":
        secondChoice(companies)
    else:
        print("Invalid selection")


main()
```

```python
def performanceBy(list, category, sort, evaluate, startDate, endDate):
    newList = []
    sum = 0
    try:
        if category == "country":
            for i in list:
                if i.getHQ().lower() == sort and i.getDate() >= startDate and i.getDate() <= endDate:
                    newList.append(i)
            for i in newList:
                sum += i.getChngD1OToD6C()
        elif category == "sector":
            for i in list:
                if i.getSector().lower() == sort and i.getDate() >= startDate and i.getDate() <= endDate:
                    newList.append(i)
            for i in newList:
                sum += i.getChngD1OToD6C()
        elif category == "industry":
            for i in list:
                if i.getIndustry().lower() == sort and i.getDate() >= startDate and i.getDate() <= endDate:
                    newList.append(i)
            for i in newList:
                sum += i.getChngD1OToD6C()
        else:
            return "Invalid Input"
        print("Average over first week: {0:6.2f}%".format((sum/len(newList)) * 100))
        print()
        header()
        for i in newList:
            printCompanyData(i)
        return
    except ZeroDivisionError:
```