**INSY 5336**

**Python Programming**

**Spring 2020**

**Final Term Project (100 points)**

**Due Date: May 7, 2020 11:59 pm CST (no exceptions)**

The following guidelines should be followed and will be used to grade your project work:

* All code to be implemented and submitted as a jupyter notebook (.ipynb) file.
* This is an individual homework assignment, no group submissions will be accepted. If you discuss in groups, please write your code individually and submit.
* Sample runs shown in the question should be used as a guide for implementation. However extensive testing needs to be done on your code to deal with all test cases that might possibly be executed.
* The instructions for running of each cell and the expected results should be documented in the cell preceding the code using markdown language.
* Every code segment in the jupyter notebook cells should be well documented with comments. Use # in the code to provide comments and they should explain the algorithm and what the code segment is doing.
* Error checking in your code is very important and differentiates a high quality programmer from a low quality one. Hence you should account for invalid user inputs, infinite loops, out of range results, etc. and resolve them by appropriate error messages. The homework will be graded for robustness of your code.
* Please read each assignment carefully. Note that you need to test your code with example input files. I will be using my own test input file to test your code. DO NOT hard code file names in your program.

This is a project to scrape data from the web and store the results in a text file.

1. (100 points) The CNN Money’s Market Movers website [(https://money.cnn.com/data/hotstocks/](https://money.cnn.com/data/hotstocks/) ) tracks the most active stocks on a real time basis. Specifically, the most active, the top gainers and top losers are listed at any instance in time. You will first write Python scripts that collect the list of most actives, gainers and losers from the above website. Next, your programs should take the ticker symbols and names of these companies (and categories) and build a csv file (called stocks.csv) with data about each stock from the website:

<https://finance.yahoo.com/quote/AMD?> , which gives the quote for ticker symbol AMD as an example.

Or, you can also use the stock info page from CNN website:

<https://money.cnn.com/quote/quote.html?symb=GE> , which gives the quote for ticker symbol GE as an example.

The data to be collected from the Yahoo Finance site should include:

OPEN price

PREV CLOSE price

VOLUME

MARKET CAP

Your code should also list the names of the companies in the order and categories listed in the website: <https://money.cnn.com/data/hotstocks/>and ask the user to choose a company to get the data on. Once the user chooses the company of interest, your program should display its corresponding data (Open, Prev Close, Volume and Market Cap).

Sample Runs (user input in RED):

This is a program to scrape data from the <https://money.cnn.com/data/hotstocks/> for a class project.

Which stock are you interested in:

Most Actives:

AMD Advanced Micro Devices Inc

GE General Electric Co

BAC Bank of America Corp

WBA Walgreens Boots Alliance Inc

AAPL Apple Inc F Ford Motor Co

FCX Freeport-McMoRan Inc

CSCO Cisco Systems Inc

OXY Occidental Petroleum Corp

MU Micron Technology Inc

Gainers:

WBA Walgreens Boots Alliance Inc

MKTX Marketaxess Holdings Inc

NVR NVR Inc

ARNC Arconic Inc

GPS Gap Inc

EQIX Equinix Inc

ULTA Ulta Beauty Inc

TTWO Take-Two Interactive Software Inc

M Macy's Inc

NWSA News Corp

Losers:

FCX Freeport-McMoRan Inc

WYNN Wynn Resorts Ltd

COTY Coty Inc

CNP CenterPoint Energy Inc

ABC AmerisourceBergen Corp

MRO Marathon Oil Corp

ATVI Activision Blizzard Inc

COG Cabot Oil & Gas Corp

XRAY Dentsply Sirona Inc

User inputs: COTY

The data for COTY Coty Inc is the following:

COTY Coty Inc

OPEN: 12.78

PREV CLOSE: 12.84

VOLUME: 2,000,995

MARKET CAP: 9.580B

The csv should look something like this for one entry:

**Losers,COTY,Coty Inc,12.78,12.84, 2,000,995, 9.580B**