

BA870 - Financial Analytics

Assignment 1

Yilun Wang / yilun830@bu.edu

In [90]:

```
# Knit
# !jupyter nbconvert -- to pdf /content/Assignment_1_B1_Yipeng_Guo.ipynb
```

```
[NbConvertApp] WARNING | pattern 'to' matched no files
[NbConvertApp] WARNING | pattern 'pdf' matched no files
[NbConvertApp] Converting notebook /content/Assignment_1_B1_Yipeng_Guo.ipynb to html
[NbConvertApp] Writing 331413 bytes to /content/Assignment_1_B1_Yipeng_Guo.html
```

Part 1 - Consolidated Statement of Operation for AAPL

In []:

```
# Install necessary Python libraries
! pip install requests
! pip install beautifulsoup4
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (2.23.0)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests) (3.0.4)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests) (1.24.3)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests) (2021.10.8)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages (4.6.3)
```

In [91]:

```
# Use the 'requests' library to download the necessary webpage from thge SEC ser  
ver.  
import requests  
  
# This header information MUST be submitted with your URL reuquest to the SEC web  
site.  
# If this header is not included or is incorrect, then you will not be abkle to  
download the data (it will be rejected by the SEC server).  
  
headers = {'Host': 'www.sec.gov', 'Connection': 'close',  
           'Accept': 'application/json, text/javascript, */*; q=0.01', 'X-Requeste  
d-With': 'XMLHttpRequest',  
           'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/53  
7.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36',  
           }  
  
URL1 = 'https://www.sec.gov/Archives/edgar/data/320193/000032019321000105/R2.ht  
m'  
  
# Get data from webpage  
# include the HEADER or you might get denied  
  
page1 = requests.get(URL1, headers=headers)
```

In [94]:

```
## Use 'Beautiful Soup' Library to implement an 'html/xml parser'  
from bs4 import BeautifulSoup  
  
soup1 = BeautifulSoup(page1.content, 'lxml')
```

In [104]:

```
# Find the tag that identify tables
Table1 = soup1.find_all('table')

# Use 'read_html' request to directly get the uncleaned table
import pandas as pd

AAPL = pd.read_html(str(Table1))[0]
AAPL
```

Out[104]:

CONSOLIDATED STATEMENTS OF OPERATIONS - USD ()shares in Thousands, in Millions		12 Months Ended		
CONSOLIDATED STATEMENTS OF OPERATIONS - USD ()shares in Thousands, in Millions		Sep. 25, 2021	Sep. 26, 2020	Sep. 28, 2019
0	Net sales	\$ 365,817	\$ 274,515	\$ 260,174
1	Cost of sales	212981	169559	161782
2	Gross margin	152836	104956	98392
3	Operating expenses:	NaN	NaN	NaN
4	Research and development	21914	18752	16217
5	Selling, general and administrative	21973	19916	18245
6	Total operating expenses	43887	38668	34462
7	Operating income	108949	66288	63930
8	Other income/(expense), net	258	803	1807
9	Income before provision for income taxes	109207	67091	65737
10	Provision for income taxes	14527	9680	10481
11	Net income	\$ 94,680	\$ 57,411	\$ 55,256
12	Earnings per share:	NaN	NaN	NaN
13	Basic (in dollars per share)	\$ 5.67	\$ 3.31	\$ 2.99
14	Diluted (in dollars per share)	\$ 5.61	\$ 3.28	\$ 2.97
15	Shares used in computing earnings per share:	NaN	NaN	NaN
16	Basic (in shares)	16701272	17352119	18471336
17	Diluted (in shares)	16864919	17528214	18595651
18	Products	NaN	NaN	NaN
19	Net sales	\$ 297,392	\$ 220,747	\$ 213,883
20	Cost of sales	192266	151286	144996
21	Services	NaN	NaN	NaN
22	Net sales	68425	53768	46291
23	Cost of sales	\$ 20,715	\$ 18,273	\$ 16,786

In [105]:

```
# Clean the dollar sign, comma sign and space.
check = AAPL.columns

AAPL[check] = AAPL[check].replace({'\$': ''}, regex = True)
AAPL[check] = AAPL[check].replace({'\,': ''}, regex = True)
AAPL[check] = AAPL[check].replace({' ': ''}, regex = True)

AAPL
```

Out[105]:

CONSOLIDATED STATEMENTS OF OPERATIONS - USD ()shares in Thousands, in Millions		12 Months Ended		
CONSOLIDATED STATEMENTS OF OPERATIONS - USD ()shares in Thousands, in Millions		Sep. 25, 2021	Sep. 26, 2020	Sep. 28, 2019
0	Netsales	365817	274515	260174
1	Cost of sales	212981	169559	161782
2	Gross margin	152836	104956	98392
3	Operating expenses:	NaN	NaN	NaN
4	Research and development	21914	18752	16217
5	Selling, general and administrative	21973	19916	18245
6	Total operating expenses	43887	38668	34462
7	Operating income	108949	66288	63930
8	Other income/(expense) net	258	803	1807
9	Income before provision for income taxes	109207	67091	65737
10	Provision for income taxes	14527	9680	10481
11	Net income	94680	57411	55256
12	Earnings per share:	NaN	NaN	NaN
13	Basic (in dollars per share)	5.67	3.31	2.99
14	Diluted (in dollars per share)	5.61	3.28	2.97
15	Shares used in computing earnings per share:	NaN	NaN	NaN
16	Basic (in shares)	16701272	17352119	18471336
17	Diluted (in shares)	16864919	17528214	18595651
18	Products	NaN	NaN	NaN
19	Netsales	297392	220747	213883
20	Cost of sales	192266	151286	144996
21	Services	NaN	NaN	NaN
22	Netsales	68425	53768	46291
23	Cost of sales	20715	18273	16786

Part 2 - Income Statement for MSFT

In [106]:

```
# Use the 'requests' library to download the necessary webpage from thge SEC ser
ver.
import requests

# This header information MUST be submitted with your URL reuquest to the SEC web
site.
# If this header is not included or is incorrect, then you will not be abkle to
download the data (it will be rejected by the SEC server).

headers = {'Host': 'www.sec.gov', 'Connection': 'close',
           'Accept': 'application/json, text/javascript, */*; q=0.01', 'X-Requeste
d-With': 'XMLHttpRequest',
           'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/53
7.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36',
           }

URL2 = 'https://www.sec.gov/Archives/edgar/data/789019/000156459021039151/R2.ht
m'

# Get data from webpage
# include the HEADER or you might get denied

page2 = requests.get(URL2, headers=headers)
```

In [107]:

```
## Use 'Beautiful Soup' Library to implement an 'html/xml parser'
from bs4 import BeautifulSoup

soup2 = BeautifulSoup(page2.content, 'lxml')
```

In [108]:

```
# Find the tag that identify tables
Table2 = soup2.find_all('table')

# Use 'read_html' request to directly get the table

MSFT = pd.read_html(str(Table2))[0]
MSFT
```

Out[108]:

INCOME STATEMENTS - USD ()sharesinMillions, in Millions		12 Months Ended		
INCOME STATEMENTS - USD ()sharesinMillions, in Millions		Jun. 30, 2021	Jun. 30, 2020	Jun. 30, 2019
0	Revenue	\$ 168,088	\$ 143,015	\$ 125,843
1	Cost of revenue	52232	46078	42910
2	Gross margin	115856	96937	82933
3	Research and development	20716	19269	16876
4	Sales and marketing	20117	19598	18213
5	General and administrative	5107	5111	4885
6	Operating income	69916	52959	42959
7	Other income, net	1186	77	729
8	Income before income taxes	71102	53036	43688
9	Provision for income taxes	9831	8755	4448
10	Net income	\$ 61,271	\$ 44,281	\$ 39,240
11	Earnings per share:	NaN	NaN	NaN
12	Basic	\$ 8.12	\$ 5.82	\$ 5.11
13	Diluted	\$ 8.05	\$ 5.76	\$ 5.06
14	Weighted average shares outstanding:	NaN	NaN	NaN
15	Basic	7547	7610	7673
16	Diluted	7608	7683	7753
17	Product	NaN	NaN	NaN
18	Revenue	\$ 71,074	\$ 68,041	\$ 66,069
19	Cost of revenue	18219	16017	16273
20	Service and Other	NaN	NaN	NaN
21	Revenue	97014	74974	59774
22	Cost of revenue	\$ 34,013	\$ 30,061	\$ 26,637

In [109]:

```
# Clean the dollar sign, comma sign and space.
check = MSFT.columns

MSFT[check] = MSFT[check].replace({'\$': ''}, regex = True)
MSFT[check] = MSFT[check].replace({'\,': ''}, regex = True)
MSFT[check] = MSFT[check].replace({' ': ''}, regex = True)

MSFT
```

Out[109]:

INCOME STATEMENTS - USD ()sharesinMillions, in Millions		12 Months Ended		
INCOME STATEMENTS - USD ()sharesinMillions, in Millions		Jun. 30, 2021	Jun. 30, 2020	Jun. 30, 2019
0	Revenue	168088	143015	125843
1	Costofrevenue	52232	46078	42910
2	Grossmargin	115856	96937	82933
3	Researchanddevelopment	20716	19269	16876
4	Salesandmarketing	20117	19598	18213
5	Generalandadministrative	5107	5111	4885
6	Operatingincome	69916	52959	42959
7	Otherincomenet	1186	77	729
8	Incomebeforeincometaxes	71102	53036	43688
9	Provisionforincometaxes	9831	8755	4448
10	Netincome	61271	44281	39240
11	Earningspershare:	NaN	NaN	NaN
12	Basic	8.12	5.82	5.11
13	Diluted	8.05	5.76	5.06
14	Weightedaveragesharesoutstanding:	NaN	NaN	NaN
15	Basic	7547	7610	7673
16	Diluted	7608	7683	7753
17	Product	NaN	NaN	NaN
18	Revenue	71074	68041	66069
19	Costofrevenue	18219	16017	16273
20	ServiceandOther	NaN	NaN	NaN
21	Revenue	97014	74974	59774
22	Costofrevenue	34013	30061	26637