DEVRY UNIVERSITY
COLLEGE OF ENGINEERING AND INFORMATION SCIENCES

Python Stock Tracking Project

By;- Temesgen Kune

Bachelor, DeVry University

Professor: Ahmed Azam

CEIS150: Programming with Objects

Jun28, 2023

INTRODUCTION

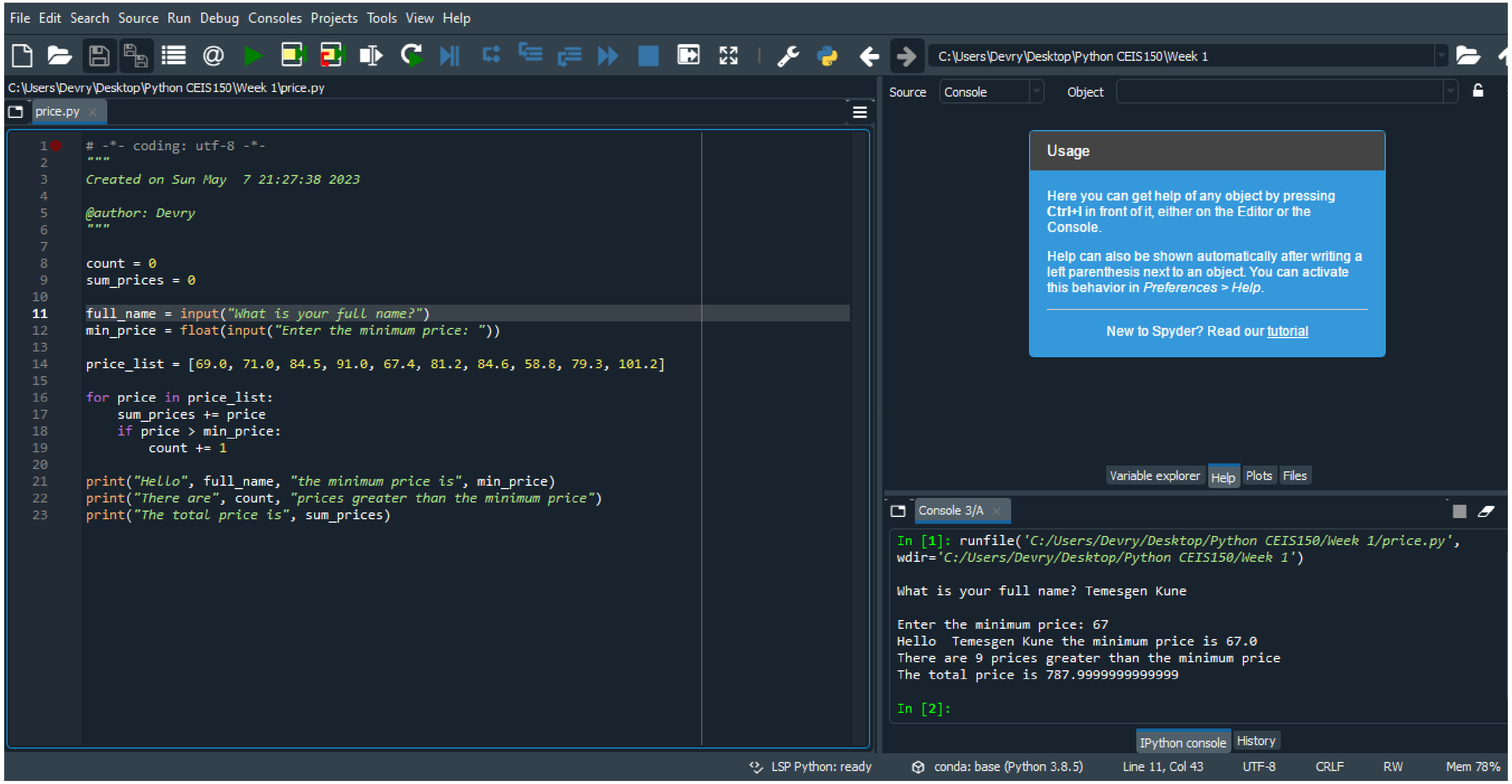
- **Creating applications in Python by using object-oriented techniques to develop a stock tracking application.**
- **The application will have both console and GUI (Graphical User Interfaces).**
- **By processing the historical stock data, profit/loss reports can be generated.**
- **The system will use the Python libraries to create charts and get historical stock data from web sites.**

Objectives – Module 1

1. Install Anaconda (if necessary)
2. Chose an IDE for your Project
3. Create a python program

Program

Screen shot of Python program running successfully.


File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Devry\Desktop\Python CEIS150\Week 1
C:\Users\Devry\Desktop\Python CEIS150\Week 1\price.py
price.py
Source Console Object
Variable explorer Help Plots Files
Console 3/A
IPython console History
LSP Python: ready conda: base (Python 3.8.5) Line 11, Col 43 UTF-8 CRLF RW Mem 78%

```python
# -*- coding: utf-8 -*-
"""
Created on Sun May  7 21:27:38 2023

@author: Devry
"""

count = 0
sum_prices = 0

full_name = input("What is your full name?")
min_price = float(input("Enter the minimum price: "))

price_list = [69.0, 71.0, 84.5, 91.0, 67.4, 81.2, 84.6, 58.8, 79.3, 101.2]

for price in price_list:
    sum_prices += price
    if price > min_price:
        count += 1

print("Hello", full_name, "the minimum price is", min_price)
print("There are", count, "prices greater than the minimum price")
print("The total price is", sum_prices)
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

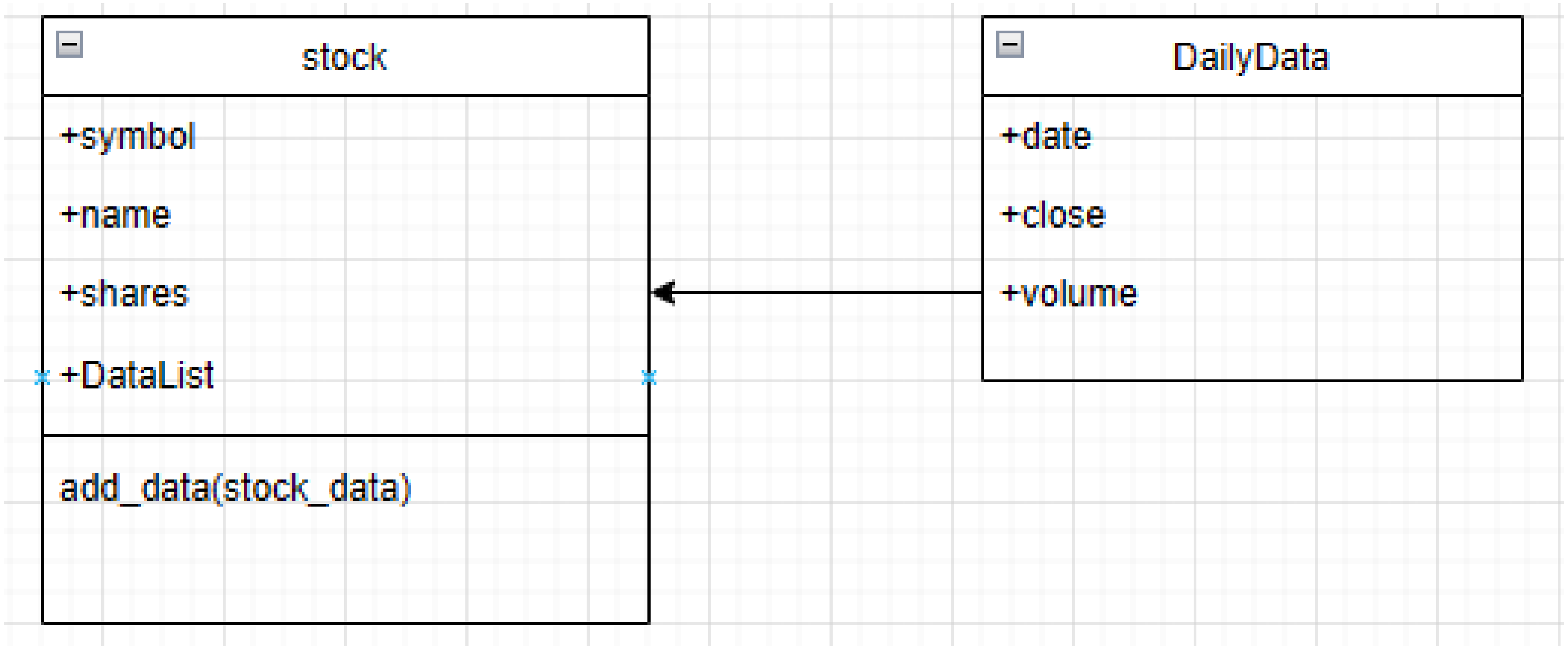
```
In [1]: runfile('C:/Users/Devry/Desktop/Python CEIS150/Week 1/price.py',
wdir='C:/Users/Devry/Desktop/Python CEIS150/Week 1')

What is your full name? Temesgen Kune

Enter the minimum price: 67
Hello  Temesgen Kune the minimum price is 67.0
There are 9 prices greater than the minimum price
The total price is 787.9999999999999

In [2]:
```

Objectives – Module 2

1. CREATE CLASS DIAGRAMS
2. CREATE CLASSES
3. RUN UNIT TEST

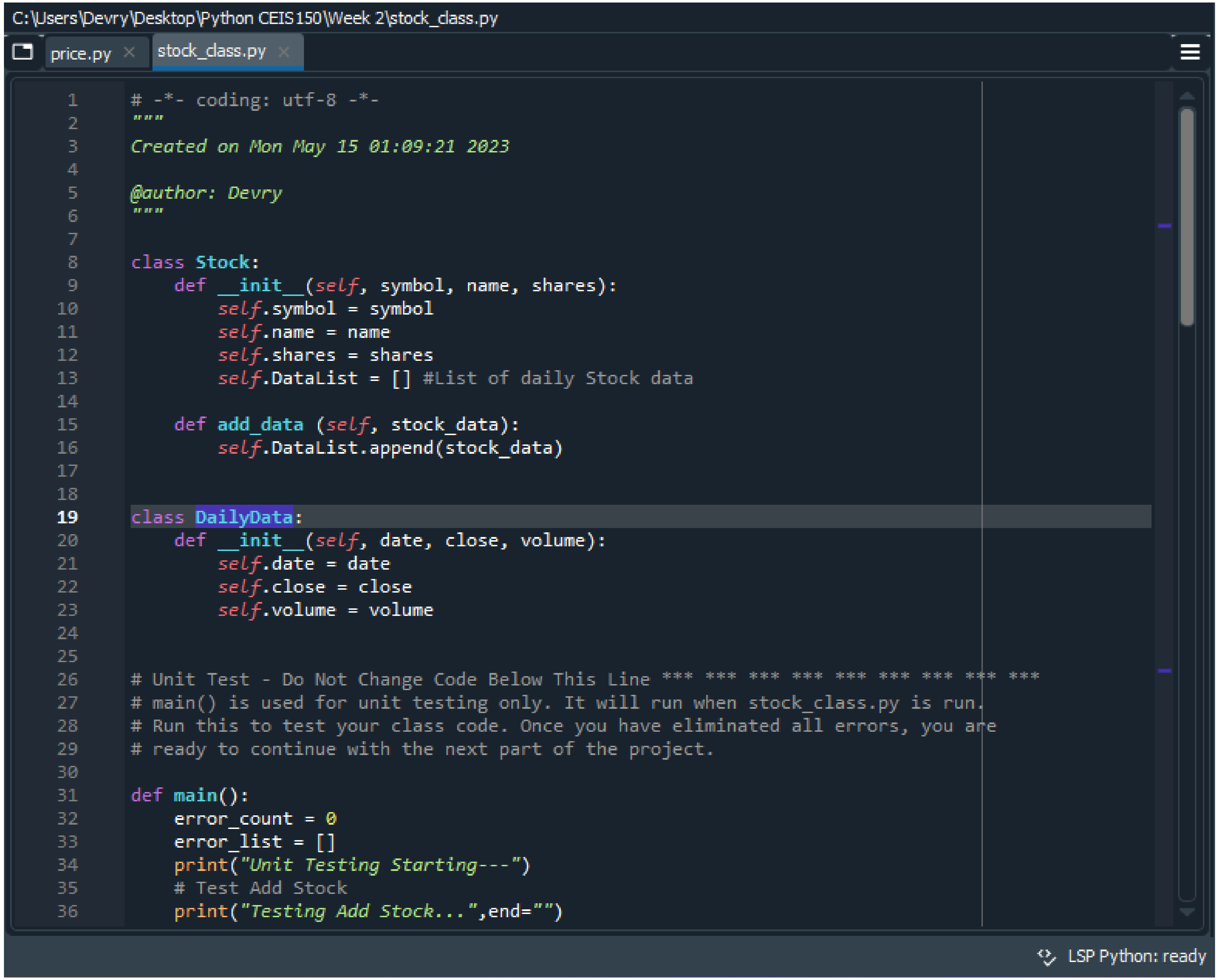
Class Diagram

Paste your Visio Class Diagram



Class Code

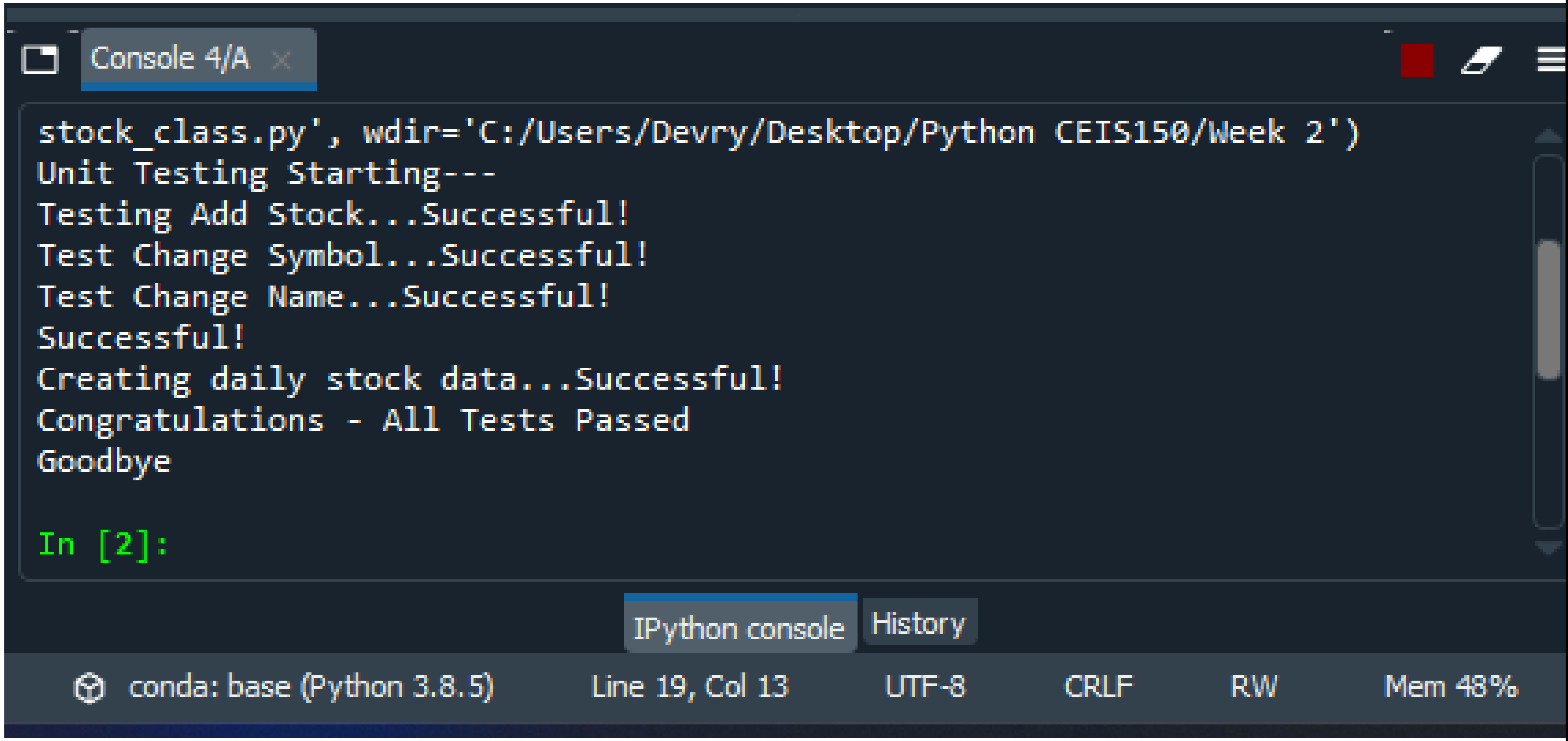
Screen Shot of your stock_class.py file.



```python
# -*- coding: utf-8 -*-
"""
Created on Mon May 15 01:09:21 2023

@author: Devry
"""

class Stock:
    def __init__(self, symbol, name, shares):
        self.symbol = symbol
        self.name = name
        self.shares = shares
        self.DataList = [] #List of daily Stock data

    def add_data (self, stock_data):
        self.DataList.append(stock_data)


class DailyData:
    def __init__(self, date, close, volume):
        self.date = date
        self.close = close
        self.volume = volume


# Unit Test - Do Not Change Code Below This Line *** *** *** *** *** *** *** *** ***
# main() is used for unit testing only. It will run when stock_class.py is run.
# Run this to test your class code. Once you have eliminated all errors, you are
# ready to continue with the next part of the project.

def main():
    error_count = 0
    error_list = []
    print("Unit Testing Starting---")
    # Test Add Stock
    print("Testing Add Stock...",end="")
```

Unit Test

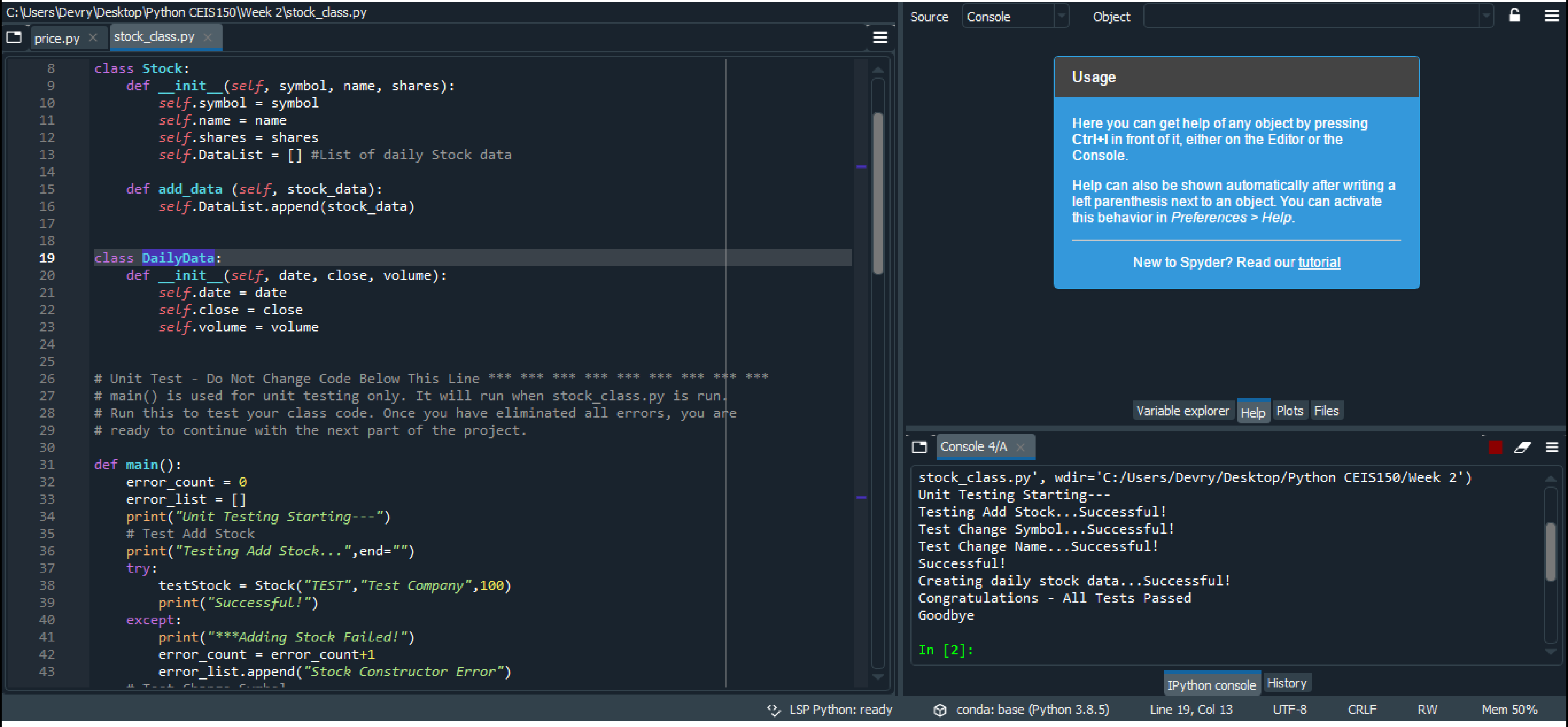
Screen Shot of your successful unit test.



```
stock_class.py', wdir='C:/Users/Devry/Desktop/Python CEIS150/Week 2')
Unit Testing Starting---
Testing Add Stock...Successful!
Test Change Symbol...Successful!
Test Change Name...Successful!
Successful!
Creating daily stock data...Successful!
Congratulations - All Tests Passed
Goodbye

In [2]:
```

Unit Test

File and success



C:\Users\Devry\Desktop\Python CEIS150\Week 2\stock_class.py

```
class Stock:
    def __init__(self, symbol, name, shares):
        self.symbol = symbol
        self.name = name
        self.shares = shares
        self.DataList = [] #List of daily Stock data

    def add_data (self, stock_data):
        self.DataList.append(stock_data)


class DailyData:
    def __init__(self, date, close, volume):
        self.date = date
        self.close = close
        self.volume = volume


# Unit Test - Do Not Change Code Below This Line *** *** *** *** *** *** *** *** ***
# main() is used for unit testing only. It will run when stock_class.py is run.
# Run this to test your class code. Once you have eliminated all errors, you are
# ready to continue with the next part of the project.

def main():
    error_count = 0
    error_list = []
    print("Unit Testing Starting---")
    # Test Add Stock
    print("Testing Add Stock...",end="")
    try:
        testStock = Stock("TEST","Test Company",100)
        print("Successful!")
    except:
        print("***Adding Stock Failed!")
        error_count = error_count+1
        error_list.append("Stock Constructor Error")
```

Usage

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
stock_class.py', wdir='C:/Users/Devry/Desktop/Python CEIS150/Week 2')
Unit Testing Starting---
Testing Add Stock...Successful!
Test Change Symbol...Successful!
Test Change Name...Successful!
Successful!
Creating daily stock data...Successful!
Congratulations - All Tests Passed
Goodbye

In [2]:
```

LSP Python: ready   conda: base (Python 3.8.5)   Line 19, Col 13   UTF-8   CRLF   RW   Mem 50%

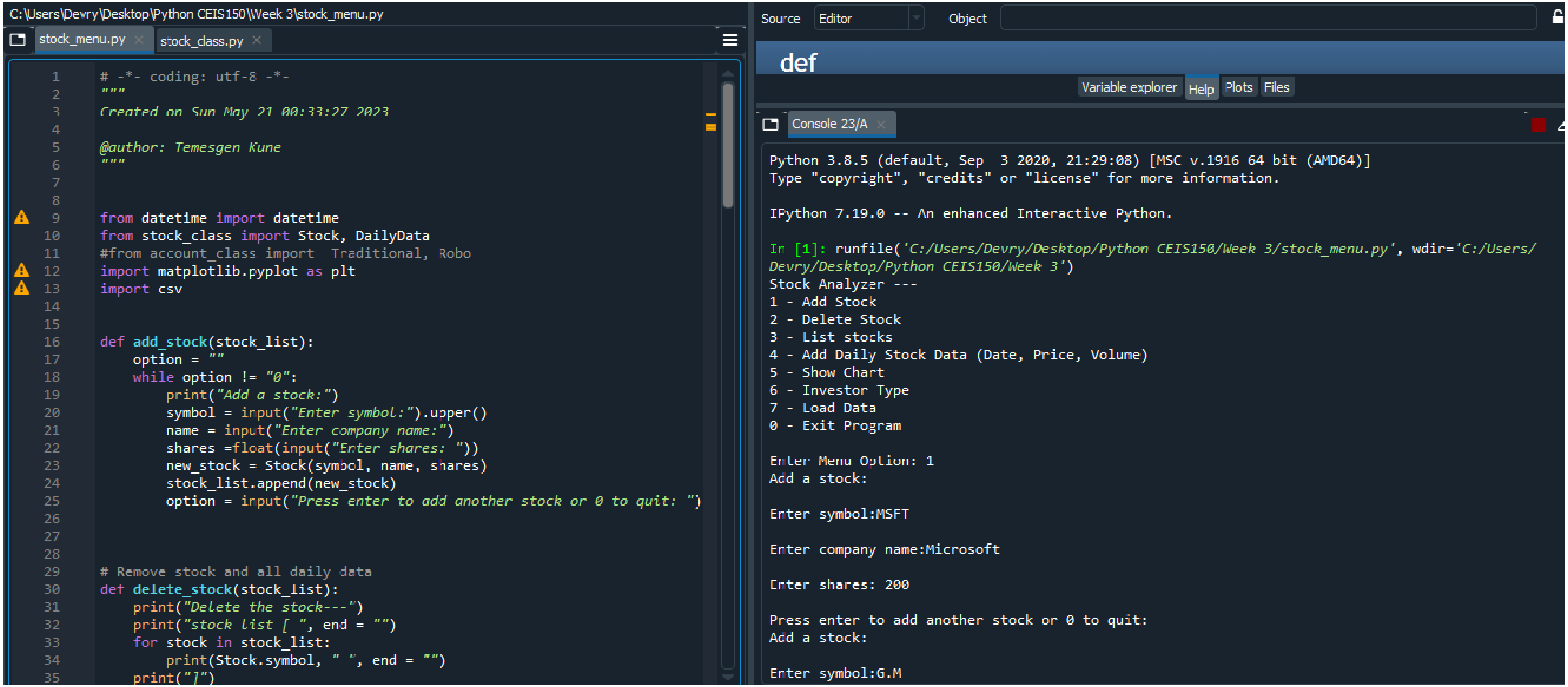
Objectives – Module 3

Create a Menu-Driven User Interface to Add/List/Delete Stocks and to Track Price and Volume History

Create Functions to handle menu items

Adding a Stock

Paste a screen shot of your working Stock program.

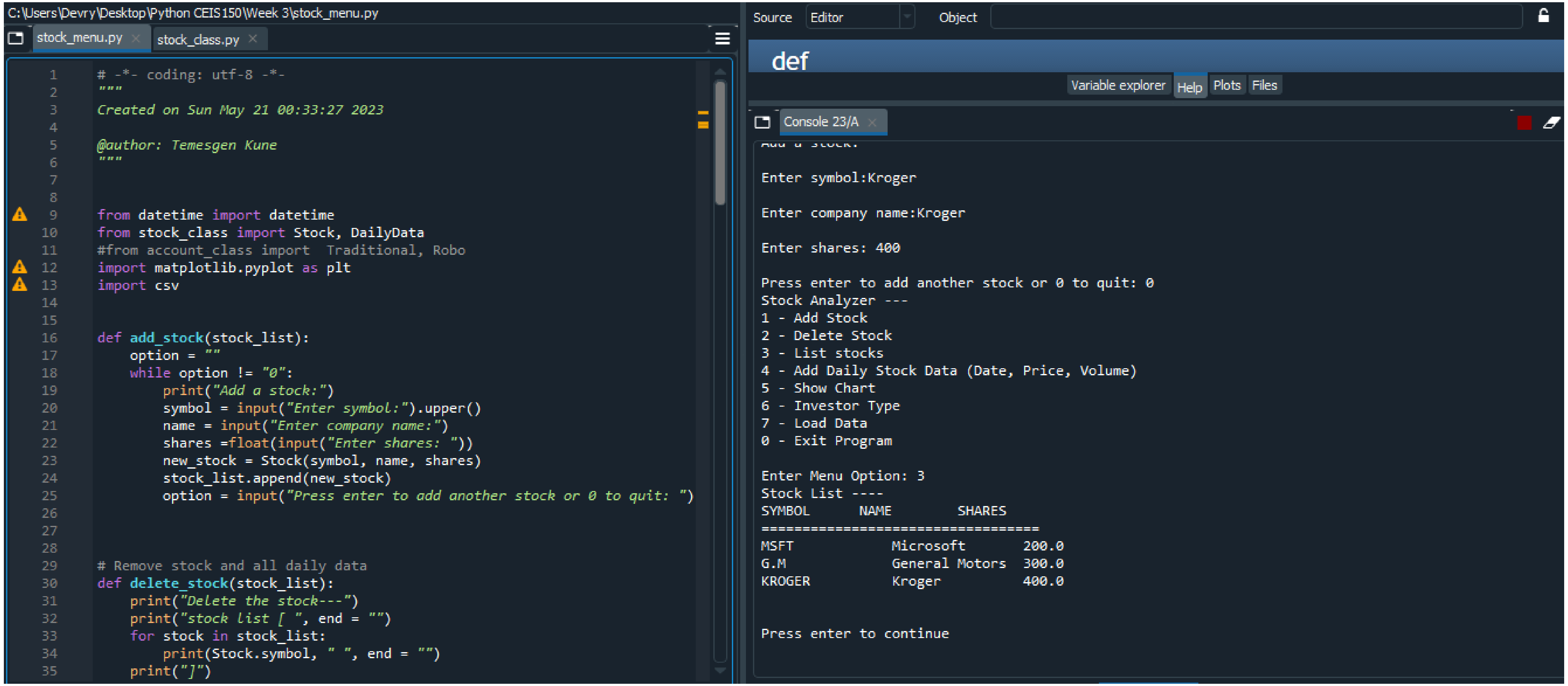


```python
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
#from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
        stock_list.append(new_stock)
        option = input("Press enter to add another stock or 0 to quit: ")



# Remove stock and all daily data
def delete_stock(stock_list):
    print("Delete the stock---")
    print("stock list [ ", end = "")
    for stock in stock_list:
        print(Stock.symbol, " ", end = "")
    print("]")
```

```
Python 3.8.5 (default, Sep  3 2020, 21:29:08) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Devry/Desktop/Python CEIS150/Week 3/stock_menu.py', wdir='C:/Users/
Devry/Desktop/Python CEIS150/Week 3')
Stock Analyzer ---
1 - Add Stock
2 - Delete Stock
3 - List stocks
4 - Add Daily Stock Data (Date, Price, Volume)
5 - Show Chart
6 - Investor Type
7 - Load Data
0 - Exit Program

Enter Menu Option: 1
Add a stock:

Enter symbol:MSFT

Enter company name:Microsoft

Enter shares: 200

Press enter to add another stock or 0 to quit:
Add a stock:

Enter symbol:G.M
```

Listing 3 Stocks

Paste a screen shot of your working Stock program.

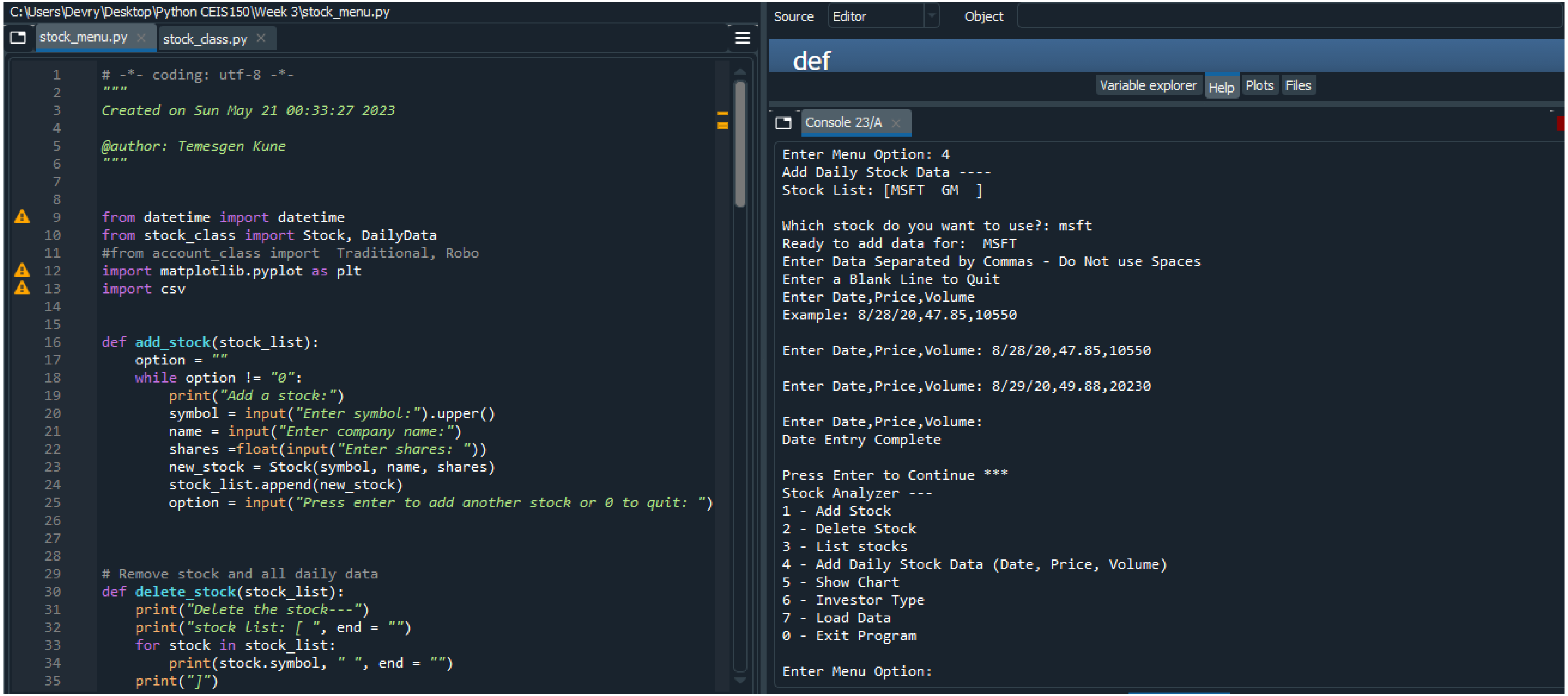


```python
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
#from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
        stock_list.append(new_stock)
        option = input("Press enter to add another stock or 0 to quit: ")



# Remove stock and all daily data
def delete_stock(stock_list):
    print("Delete the stock---")
    print("stock list [ ", end = "")
    for stock in stock_list:
        print(Stock.symbol, " ", end = "")
    print("]")
```

```
Enter symbol:Kroger

Enter company name:Kroger

Enter shares: 400

Press enter to add another stock or 0 to quit: 0
Stock Analyzer ---
1 - Add Stock
2 - Delete Stock
3 - List stocks
4 - Add Daily Stock Data (Date, Price, Volume)
5 - Show Chart
6 - Investor Type
7 - Load Data
0 - Exit Program

Enter Menu Option: 3
Stock List ----
SYMBOL      NAME        SHARES
==================================
MSFT            Microsoft       200.0
G.M             General Motors  300.0
KROGER          Kroger          400.0


Press enter to continue
```

Daily Data

Paste a screen shot of your working Stock program.

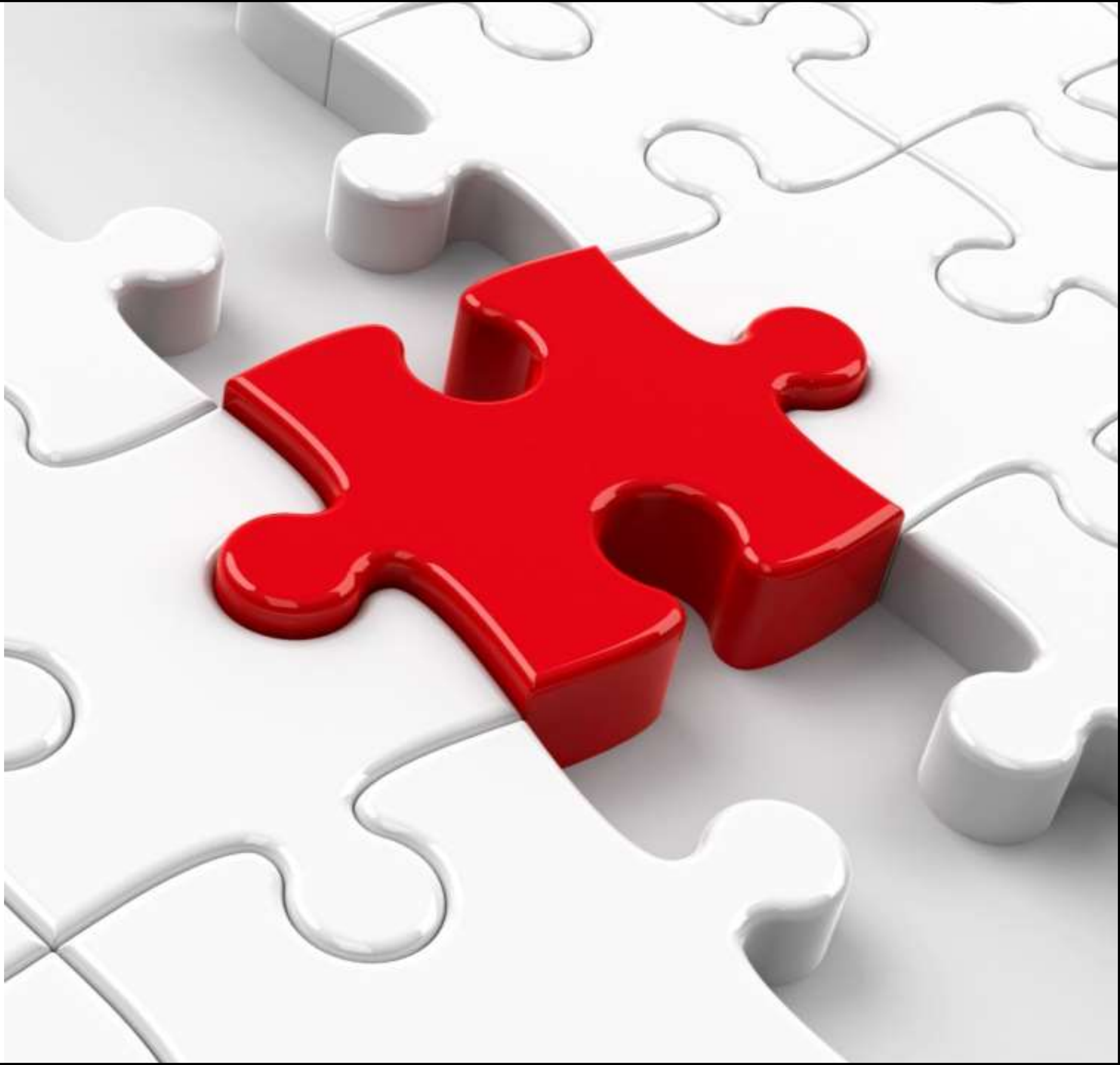


```python
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
#from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
        stock_list.append(new_stock)
        option = input("Press enter to add another stock or 0 to quit: ")



# Remove stock and all daily data
def delete_stock(stock_list):
    print("Delete the stock---")
    print("stock list: [ ", end = "")
    for stock in stock_list:
        print(stock.symbol, " ", end = "")
    print("]")
```

```
Enter Menu Option: 4
Add Daily Stock Data ----
Stock List: [MSFT   GM   ]

Which stock do you want to use?: msft
Ready to add data for:  MSFT
Enter Data Separated by Commas - Do Not use Spaces
Enter a Blank Line to Quit
Enter Date,Price,Volume
Example: 8/28/20,47.85,10550

Enter Date,Price,Volume: 8/28/20,47.85,10550

Enter Date,Price,Volume: 8/29/20,49.88,20230

Enter Date,Price,Volume:
Date Entry Complete

Press Enter to Continue ***
Stock Analyzer ---
1 - Add Stock
2 - Delete Stock
3 - List stocks
4 - Add Daily Stock Data (Date, Price, Volume)
5 - Show Chart
6 - Investor Type
7 - Load Data
0 - Exit Program

Enter Menu Option:
```

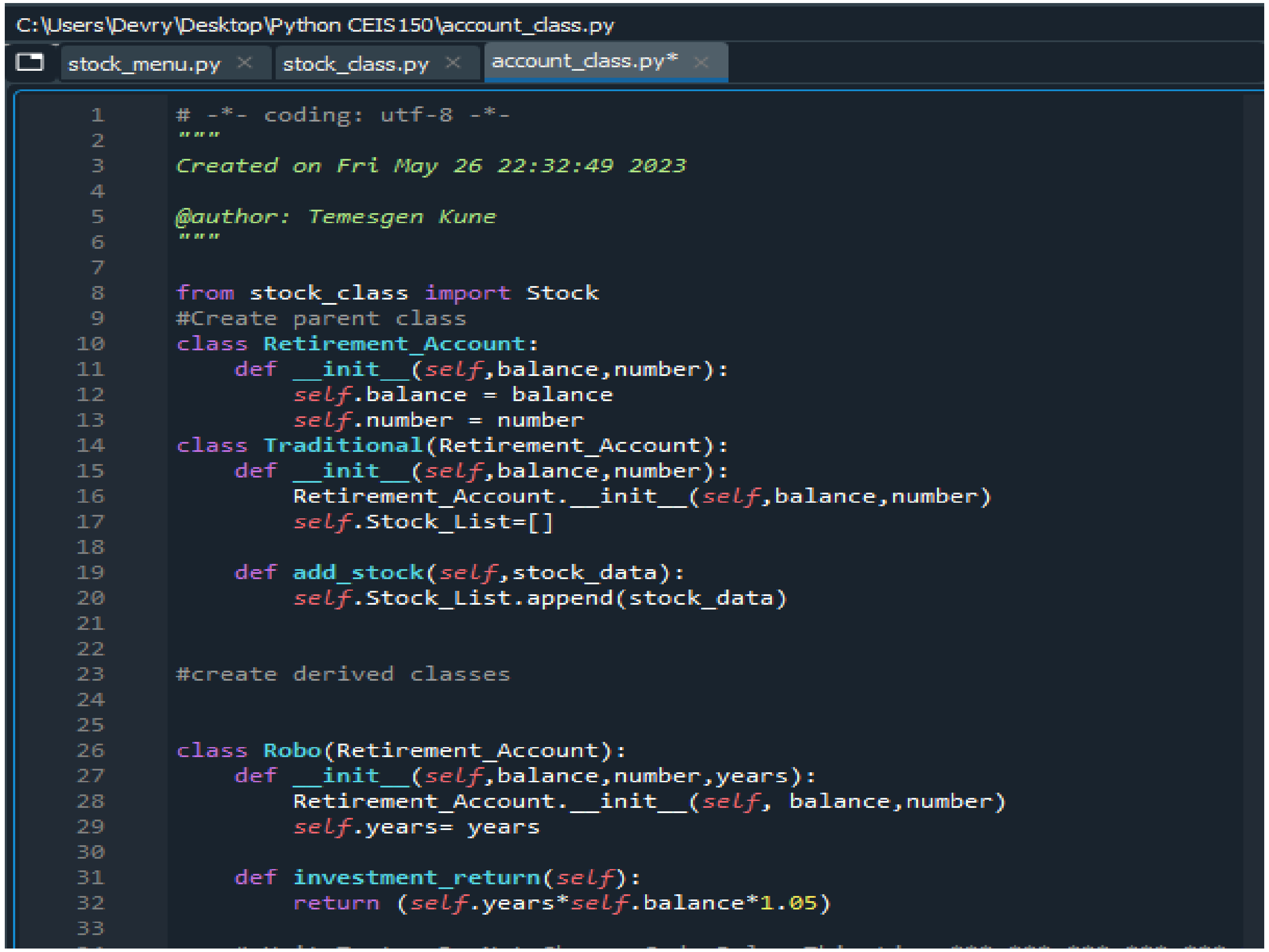
Objectives – Module 4

1. Implement inheritance in the stock program
2. Create three classes
3. Run Unit Test



Inherited classes

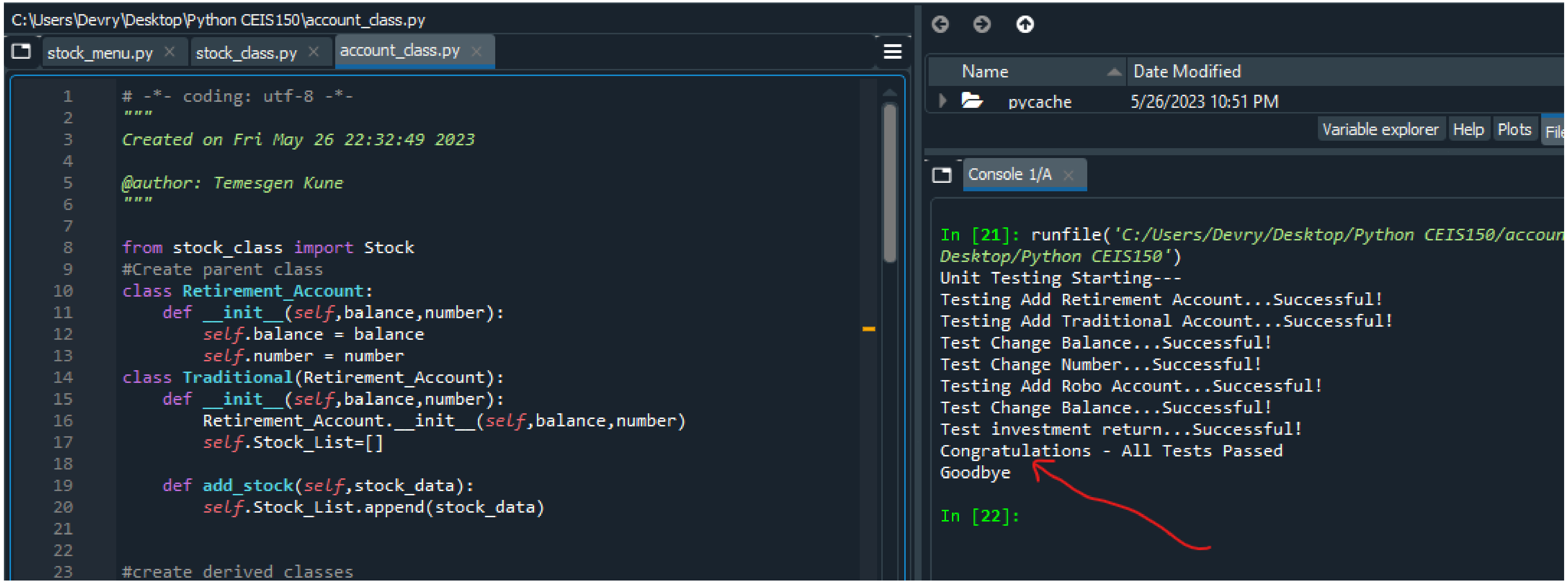
- Paste a screen shot of your classes



```python
# -*- coding: utf-8 -*-
"""
Created on Fri May 26 22:32:49 2023

@author: Temesgen Kune
"""

from stock_class import Stock
#Create parent class
class Retirement_Account:
    def __init__(self,balance,number):
        self.balance = balance
        self.number = number
class Traditional(Retirement_Account):
    def __init__(self,balance,number):
        Retirement_Account.__init__(self,balance,number)
        self.Stock_List=[]

    def add_stock(self,stock_data):
        self.Stock_List.append(stock_data)


#create derived classes


class Robo(Retirement_Account):
    def __init__(self,balance,number,years):
        Retirement_Account.__init__(self, balance,number)
        self.years= years

    def investment_return(self):
        return (self.years*self.balance*1.05)

```

Unit Tests

- Paste a screen shot of your unit tests successfully completed



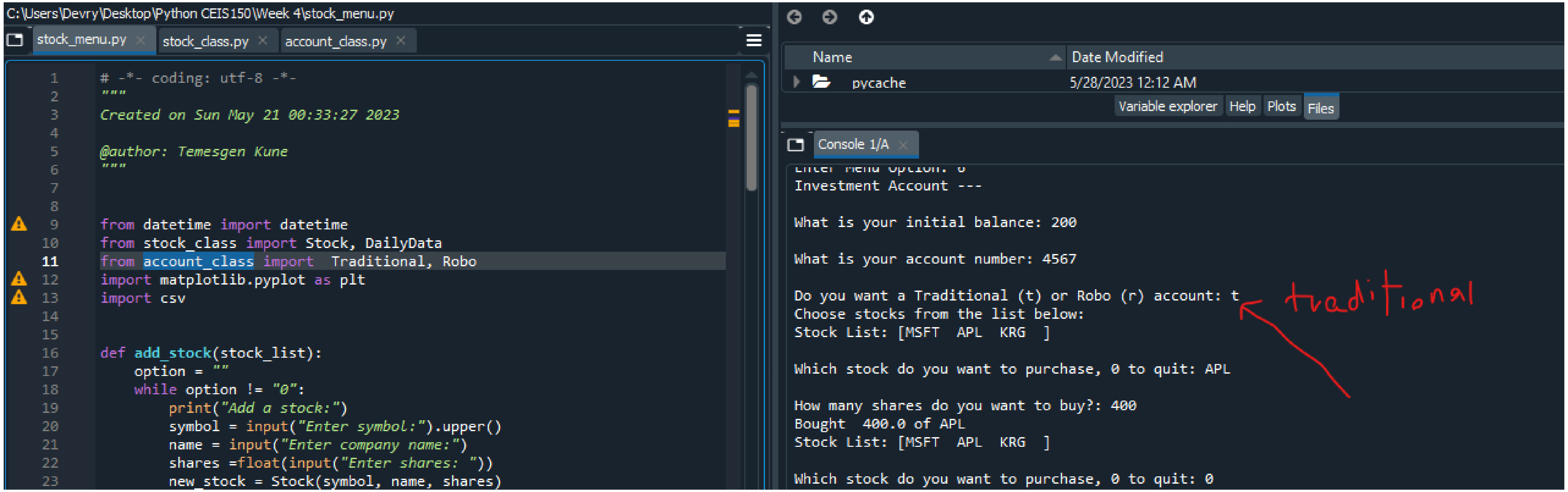
C:\Users\Devry\Desktop\Python CEIS150\account_class.py

```python
# -*- coding: utf-8 -*-
"""
Created on Fri May 26 22:32:49 2023

@author: Temesgen Kune
"""

from stock_class import Stock
#Create parent class
class Retirement_Account:
    def __init__(self,balance,number):
        self.balance = balance
        self.number = number
class Traditional(Retirement_Account):
    def __init__(self,balance,number):
        Retirement_Account.__init__(self,balance,number)
        self.Stock_List=[]

    def add_stock(self,stock_data):
        self.Stock_List.append(stock_data)


#create derived classes
```

```
In [21]: runfile('C:/Users/Devry/Desktop/Python CEIS150/accoun
Desktop/Python CEIS150')
Unit Testing Starting---
Testing Add Retirement Account...Successful!
Testing Add Traditional Account...Successful!
Test Change Balance...Successful!
Test Change Number...Successful!
Testing Add Robo Account...Successful!
Test Change Balance...Successful!
Test investment return...Successful!
Congratulations - All Tests Passed
Goodbye

In [22]:
```

Stock menu program

- Paste a screen shot of your classes in the main program
- Traditional Type

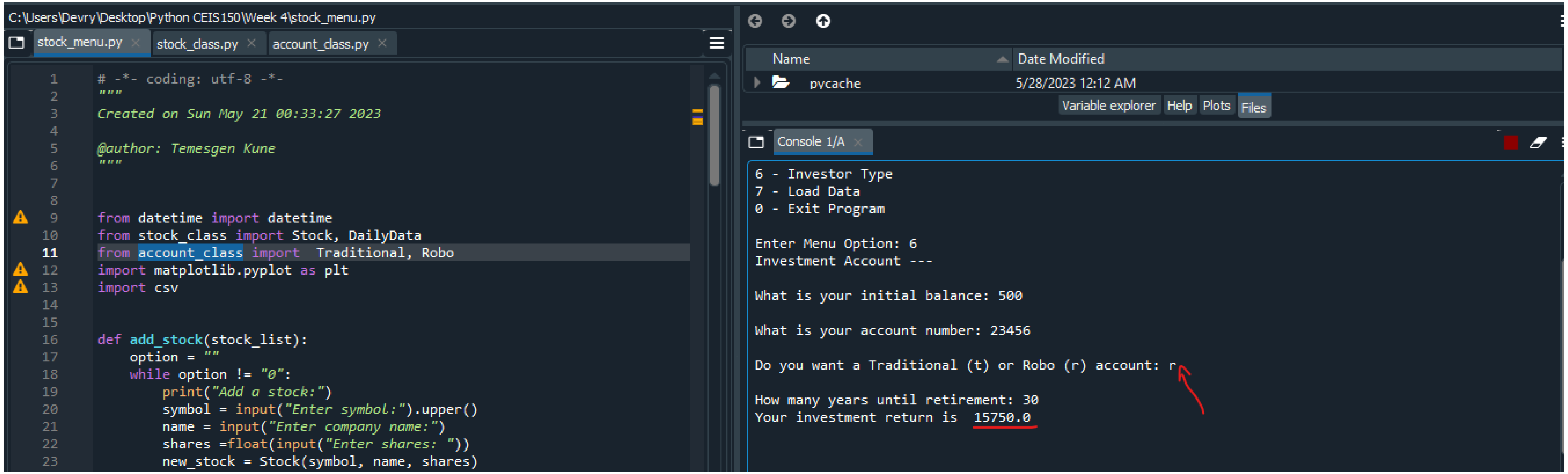


```python
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
```

```
Investment Account ---

What is your initial balance: 200

What is your account number: 4567

Do you want a Traditional (t) or Robo (r) account: t
Choose stocks from the list below:
Stock List: [MSFT  APL  KRG  ]

Which stock do you want to purchase, 0 to quit: APL

How many shares do you want to buy?: 400
Bought  400.0 of APL
Stock List: [MSFT  APL  KRG  ]

Which stock do you want to purchase, 0 to quit: 0
```

Stock menu program

- Paste a screen shot of your classes in the main program
- Robo type



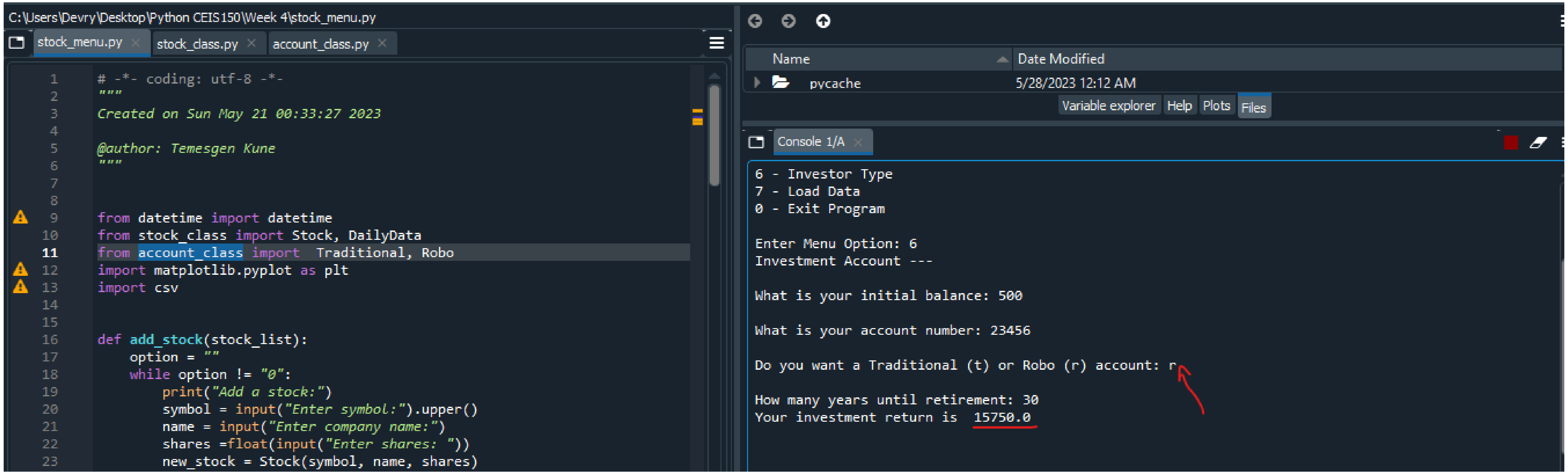
C:\Users\Devry\Desktop\Python CEIS150\Week 4\stock_menu.py

```
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
```

```
6 - Investor Type
7 - Load Data
0 - Exit Program

Enter Menu Option: 6
Investment Account ---

What is your initial balance: 500

What is your account number: 23456

Do you want a Traditional (t) or Robo (r) account: r

How many years until retirement: 30
Your investment return is  15750.0
```

Stock menu program

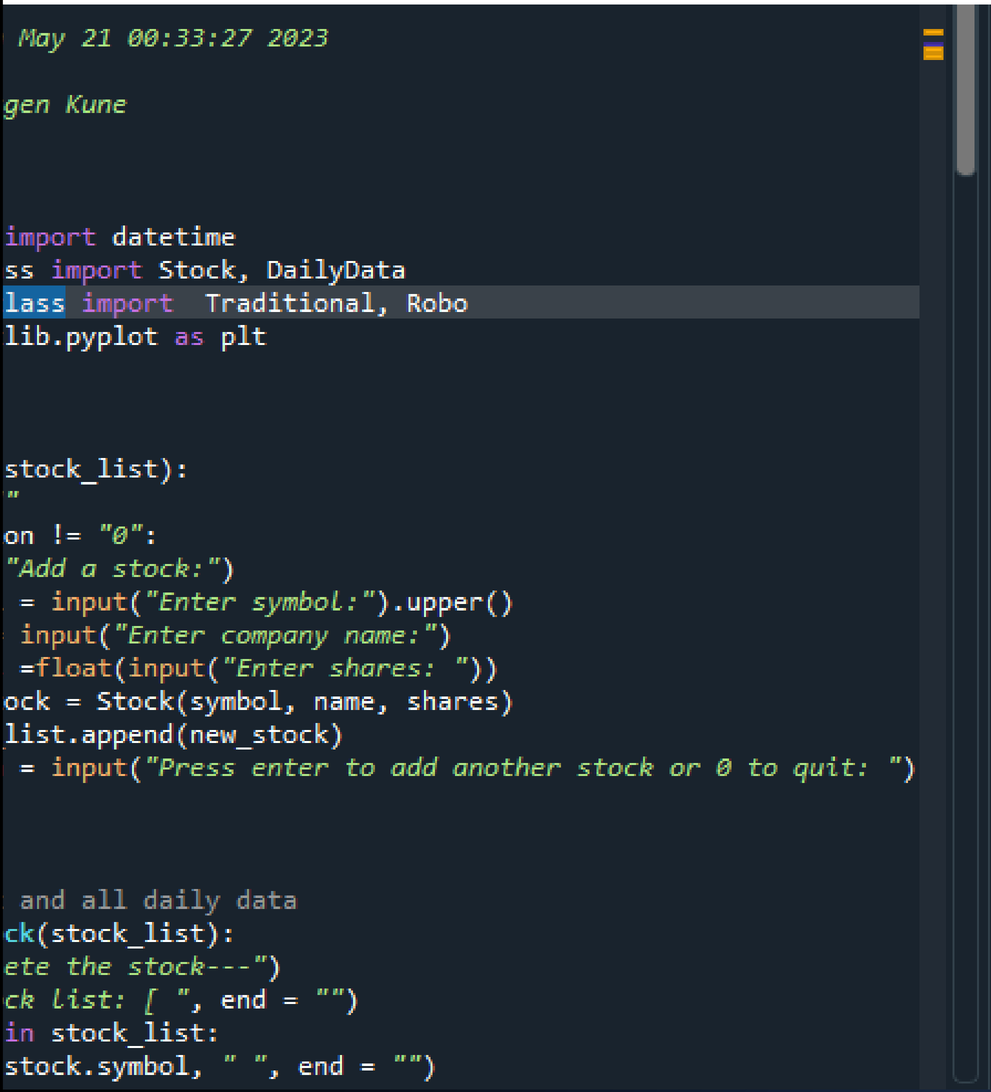
- Paste a screen shot of your classes in the main program
- Robo type



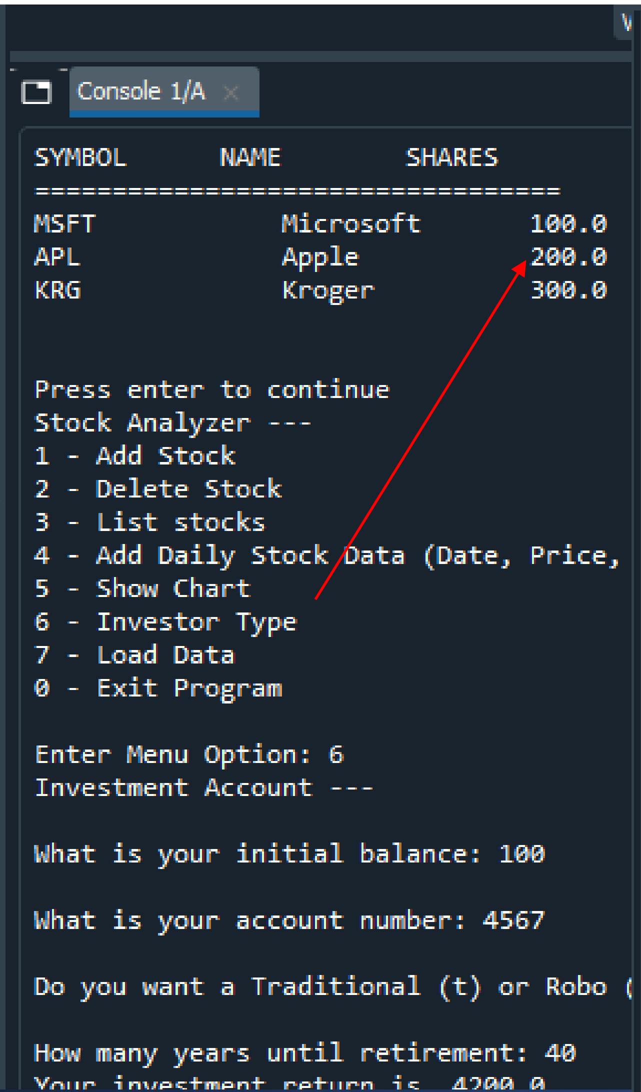
C:\Users\Devry\Desktop\Python CEIS150\Week 4\stock_menu.py

```python
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
```

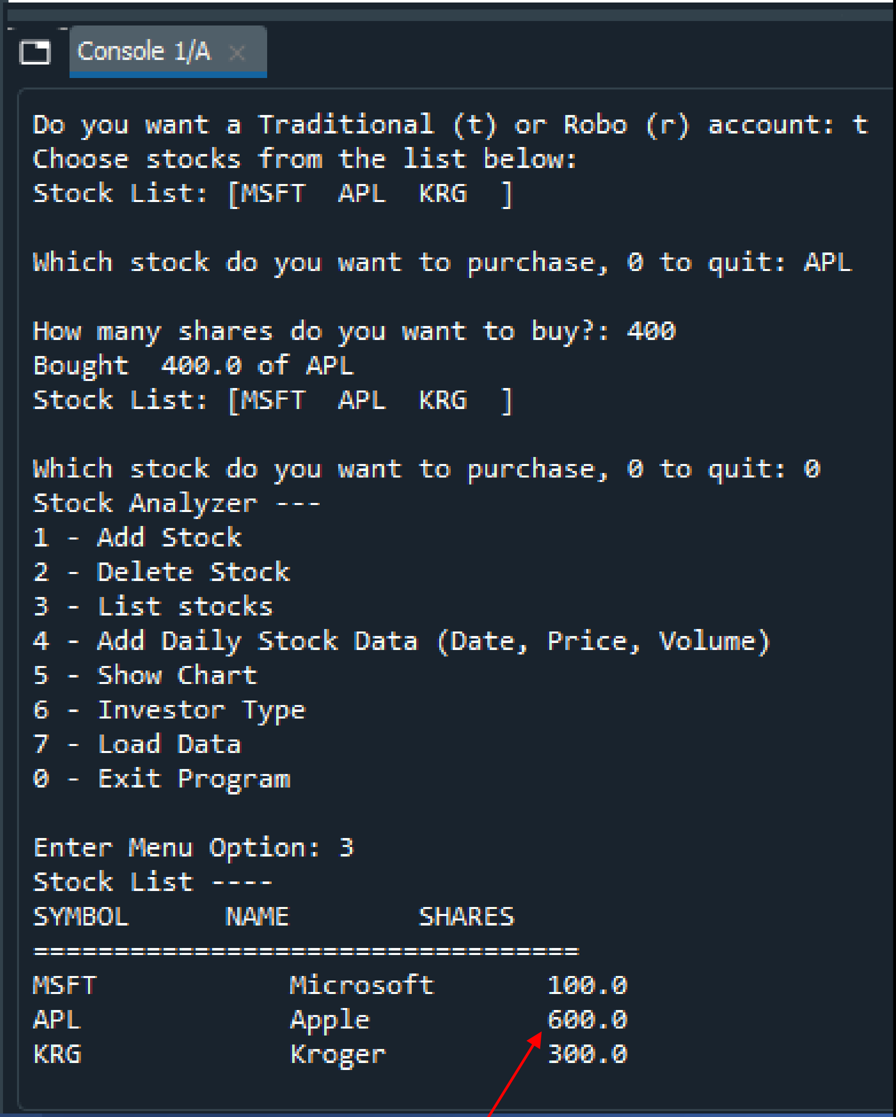
```
6 - Investor Type
7 - Load Data
0 - Exit Program

Enter Menu Option: 6
Investment Account ---

What is your initial balance: 500

What is your account number: 23456

Do you want a Traditional (t) or Robo (r) account: r

How many years until retirement: 30
Your investment return is  15750.0
```



```
May 21 00:33:27 2023

gen Kune


import datetime
ss import Stock, DailyData
lass import  Traditional, Robo
lib.pyplot as plt



stock_list):
"
on != "0":
"Add a stock:")
 = input("Enter symbol:").upper()
 input("Enter company name:")
 =float(input("Enter shares: "))
ock = Stock(symbol, name, shares)
list.append(new_stock)
 = input("Press enter to add another stock or 0 to quit: ")



 and all daily data
ck(stock_list):
ete the stock---")
ck list: [ ", end = "")
in stock_list:
stock.symbol, " ", end = "")
```



```
SYMBOL       NAME         SHARES
==================================
MSFT             Microsoft        100.0
APL              Apple            200.0
KRG              Kroger           300.0


Press enter to continue
Stock Analyzer ---
1 - Add Stock
2 - Delete Stock
3 - List stocks
4 - Add Daily Stock Data (Date, Price,
5 - Show Chart
6 - Investor Type
7 - Load Data
0 - Exit Program

Enter Menu Option: 6
Investment Account ---

What is your initial balance: 100

What is your account number: 4567

Do you want a Traditional (t) or Robo (

How many years until retirement: 40
Your investment return is  4200.0
```



```
Do you want a Traditional (t) or Robo (r) account: t
Choose stocks from the list below:
Stock List: [MSFT  APL  KRG  ]

Which stock do you want to purchase, 0 to quit: APL

How many shares do you want to buy?: 400
Bought  400.0 of APL
Stock List: [MSFT  APL  KRG  ]

Which stock do you want to purchase, 0 to quit: 0
Stock Analyzer ---
1 - Add Stock
2 - Delete Stock
3 - List stocks
4 - Add Daily Stock Data (Date, Price, Volume)
5 - Show Chart
6 - Investor Type
7 - Load Data
0 - Exit Program

Enter Menu Option: 3
Stock List ----
SYMBOL       NAME         SHARES
==================================
MSFT             Microsoft        100.0
APL              Apple            600.0
KRG              Kroger           300.0
```

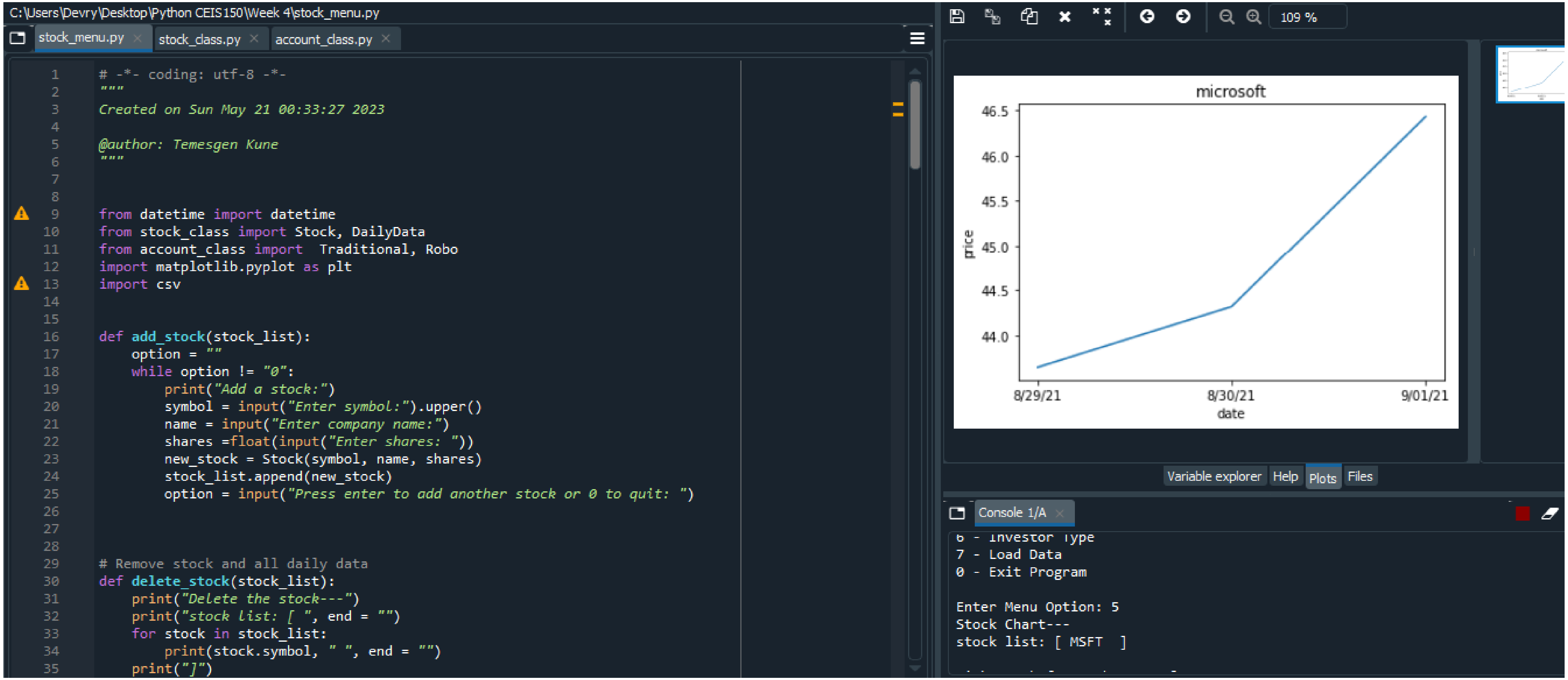
Additional Slides (After Traditional shares)

Objectives – Module 5

Use the pyplot class in the matplotlib library to create a simple stock chart.

Chart

Paste a screen shot of your stock chart.



```python
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
        stock_list.append(new_stock)
        option = input("Press enter to add another stock or 0 to quit: ")



# Remove stock and all daily data
def delete_stock(stock_list):
    print("Delete the stock---")
    print("stock list: [ ", end = "")
    for stock in stock_list:
        print(stock.symbol, " ", end = "")
    print("]")
```

```
6 - Investor Type
7 - Load Data
0 - Exit Program

Enter Menu Option: 5
Stock Chart---
stock list: [ MSFT  ]
```

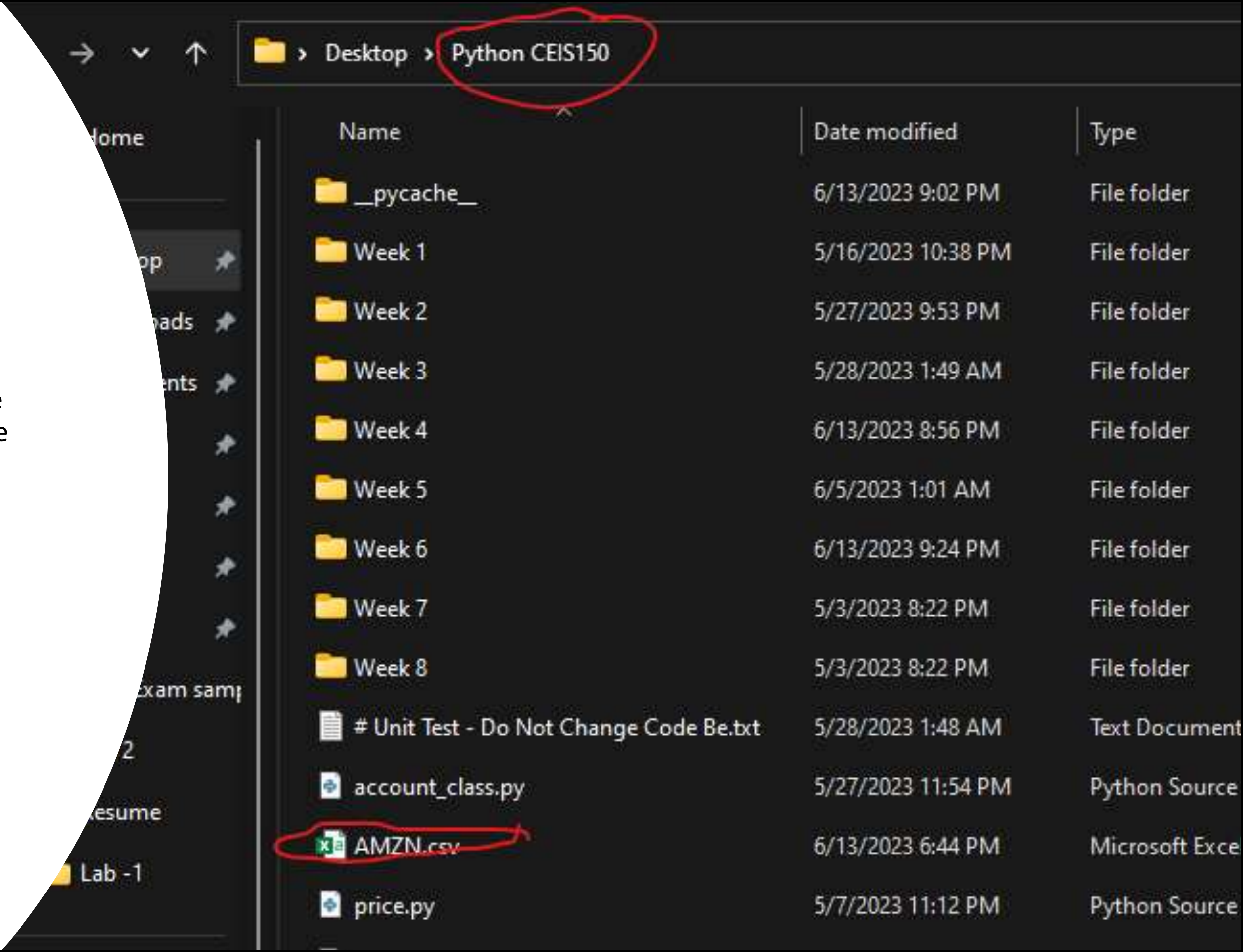
Objectives – Module 6

Use the csv library to import data from Yahoo! Finance.

Read data from a file

File

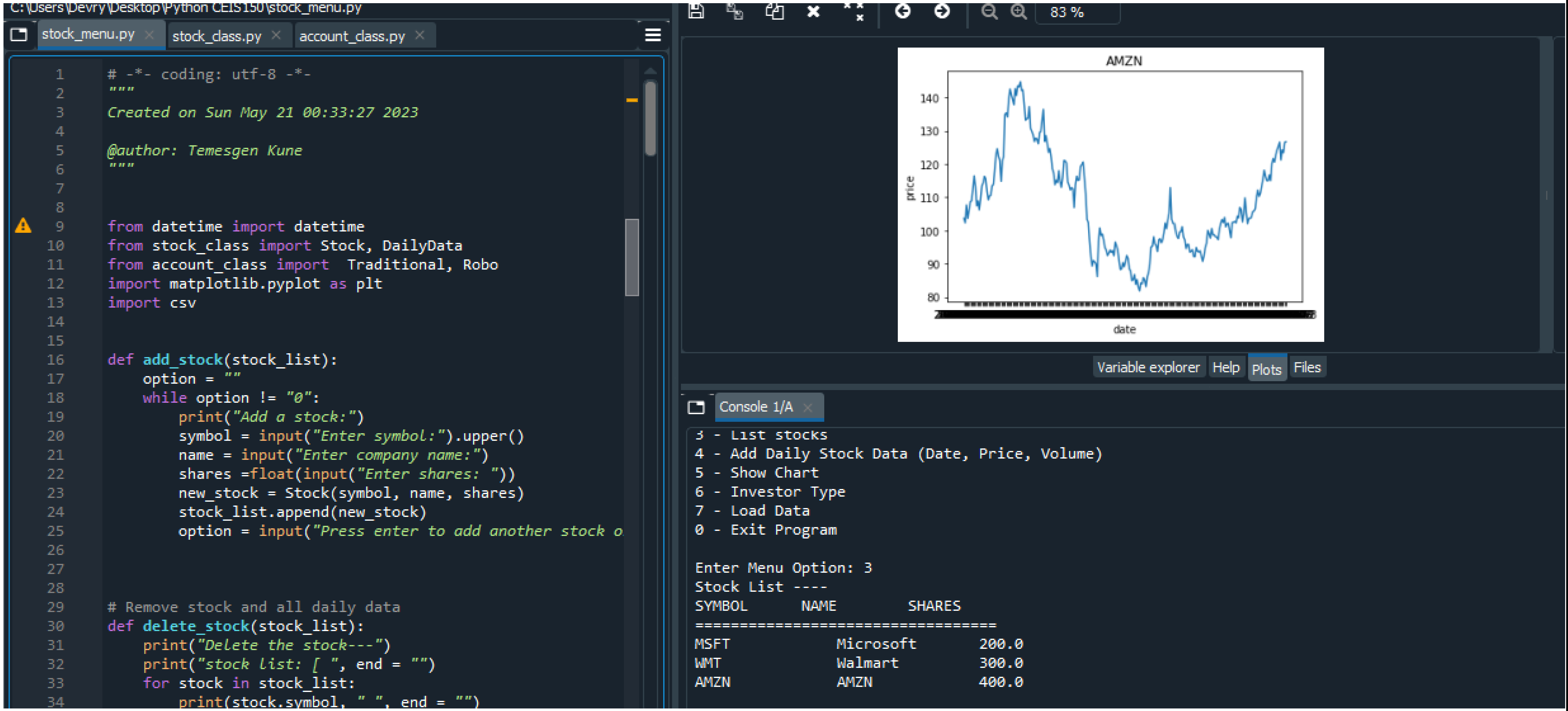
- Paste a screen shot of the file downloaded from Yahoo finance



Desktop › Python CEIS150

Name	Date modified	Type
__pycache__	6/13/2023 9:02 PM	File folder
Week 1	5/16/2023 10:38 PM	File folder
Week 2	5/27/2023 9:53 PM	File folder
Week 3	5/28/2023 1:49 AM	File folder
Week 4	6/13/2023 8:56 PM	File folder
Week 5	6/5/2023 1:01 AM	File folder
Week 6	6/13/2023 9:24 PM	File folder
Week 7	5/3/2023 8:22 PM	File folder
Week 8	5/3/2023 8:22 PM	File folder
# Unit Test - Do Not Change Code Be.txt	5/28/2023 1:48 AM	Text Document
account_class.py	5/27/2023 11:54 PM	Python Source
AMZN.csv	6/13/2023 6:44 PM	Microsoft Exce
price.py	5/7/2023 11:12 PM	Python Source

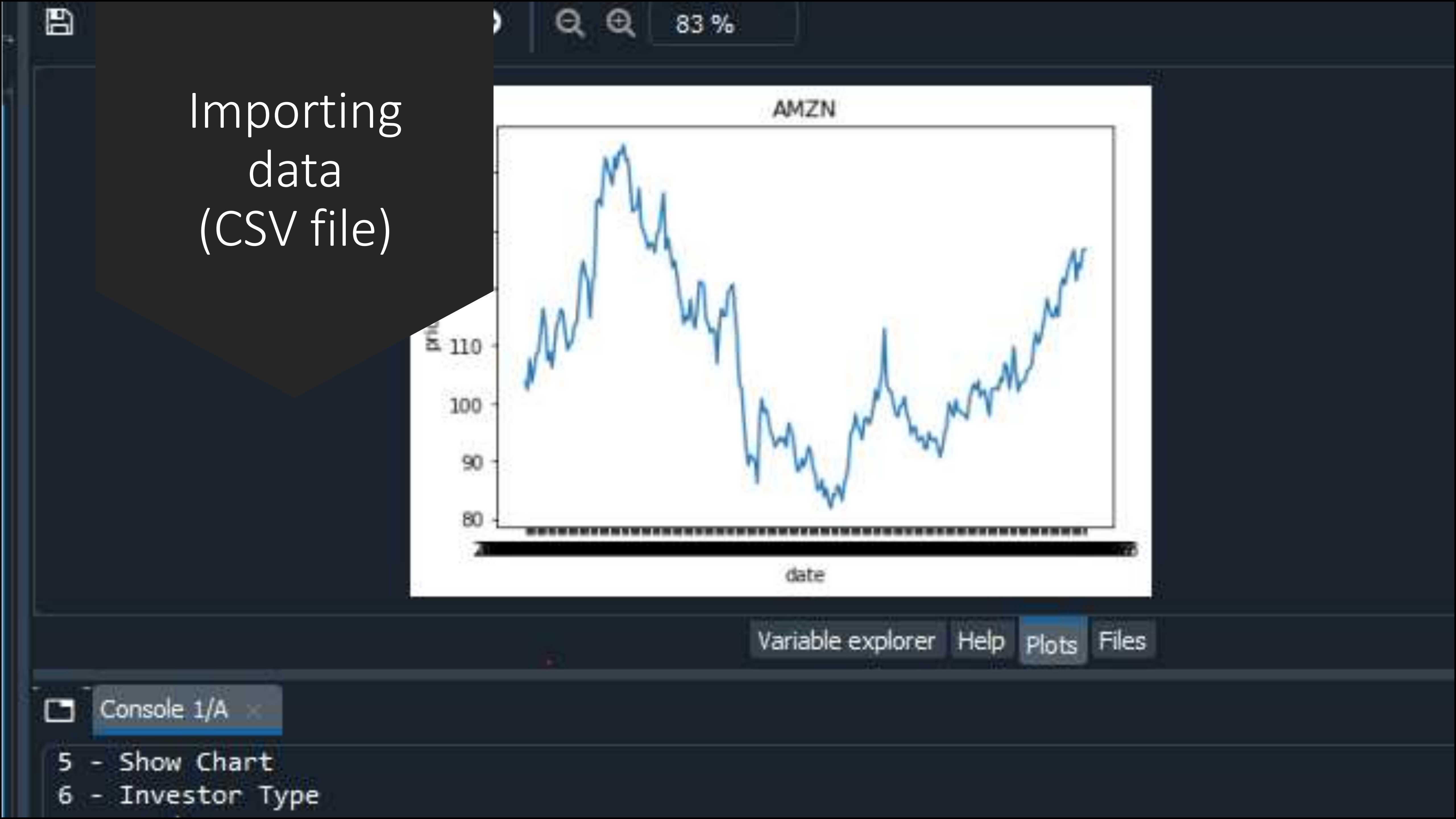
Stock Lists



```python
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
        stock_list.append(new_stock)
        option = input("Press enter to add another stock o



# Remove stock and all daily data
def delete_stock(stock_list):
    print("Delete the stock---")
    print("stock list: [ ", end = "")
    for stock in stock_list:
        print(stock.symbol, " ", end = "")
```

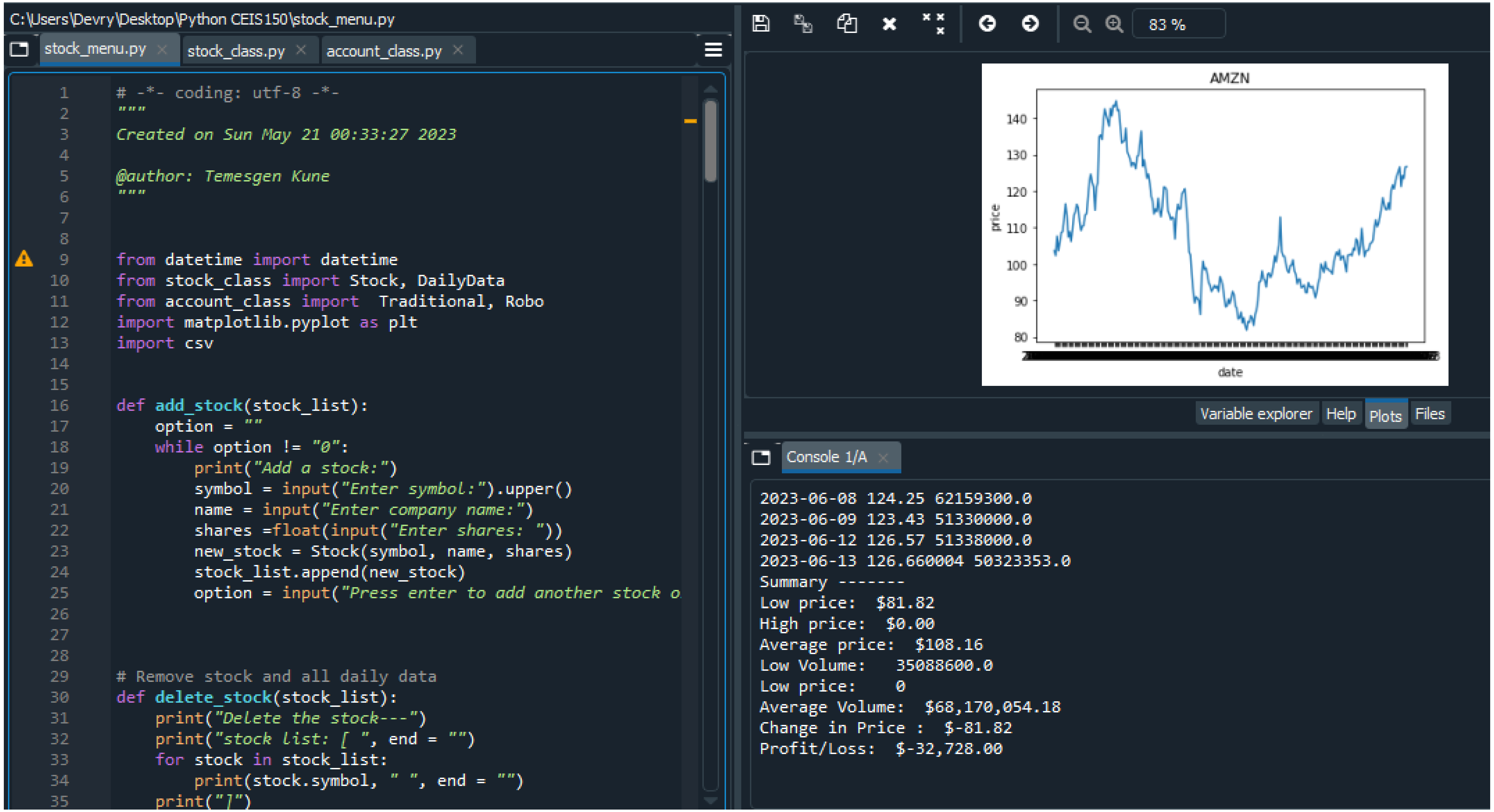
```
3 - List stocks
4 - Add Daily Stock Data (Date, Price, Volume)
5 - Show Chart
6 - Investor Type
7 - Load Data
0 - Exit Program

Enter Menu Option: 3
Stock List ----
SYMBOL      NAME        SHARES
==================================
MSFT            Microsoft       200.0
WMT             Walmart         300.0
AMZN            AMZN            400.0
```

Importing data (CSV file)



```
5 - Show Chart
6 - Investor Type
```

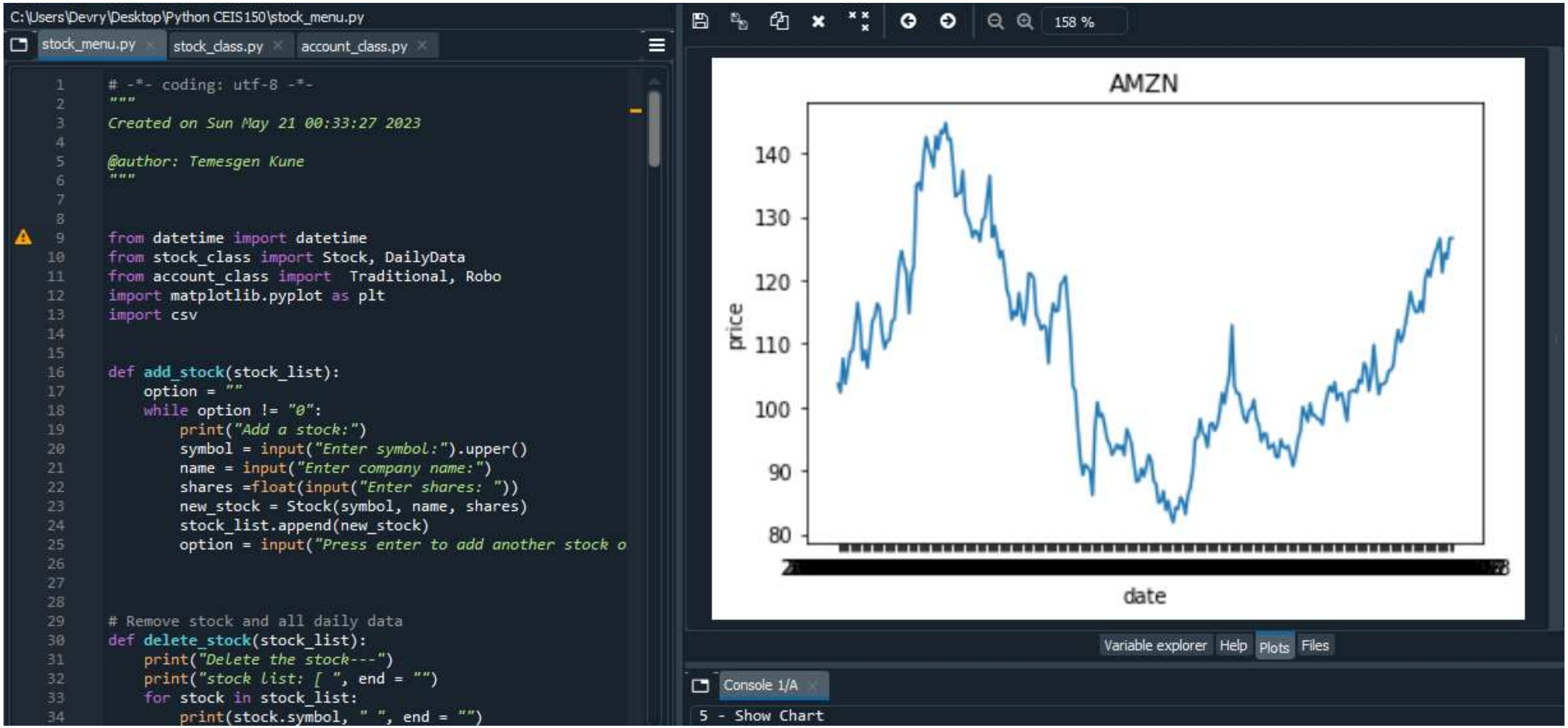
Statistical Data



```python
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
        stock_list.append(new_stock)
        option = input("Press enter to add another stock o



# Remove stock and all daily data
def delete_stock(stock_list):
    print("Delete the stock---")
    print("stock list: [ ", end = "")
    for stock in stock_list:
        print(stock.symbol, " ", end = "")
    print("]")
```

```
2023-06-08 124.25 62159300.0
2023-06-09 123.43 51330000.0
2023-06-12 126.57 51338000.0
2023-06-13 126.660004 50323353.0
Summary -------
Low price:  $81.82
High price:  $0.00
Average price:  $108.16
Low Volume:   35088600.0
Low price:    0
Average Volume:  $68,170,054.18
Change in Price :  $-81.82
Profit/Loss:  $-32,728.00
```

Chart

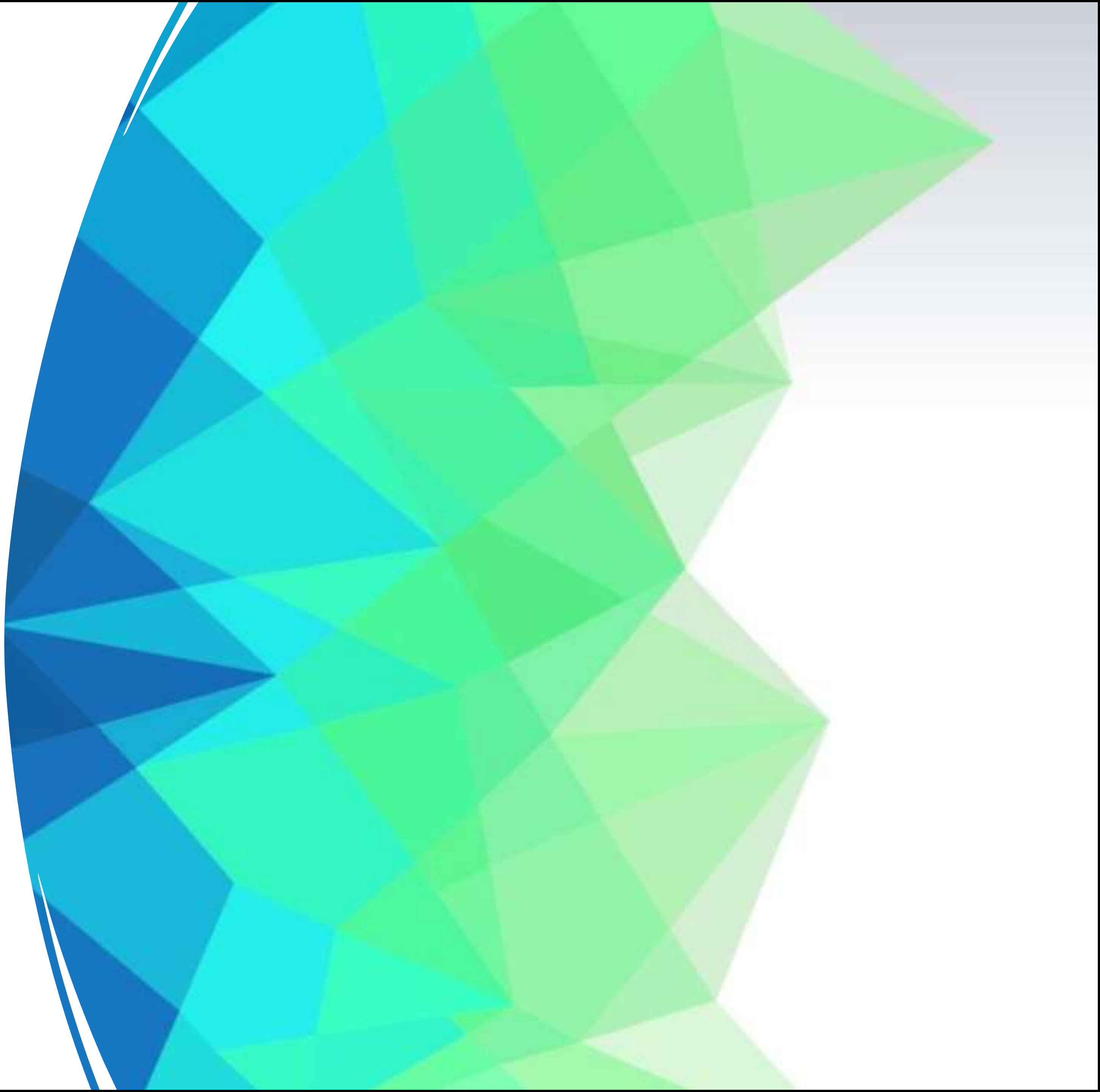


```python
# -*- coding: utf-8 -*-
"""
Created on Sun May 21 00:33:27 2023

@author: Temesgen Kune
"""


from datetime import datetime
from stock_class import Stock, DailyData
from account_class import  Traditional, Robo
import matplotlib.pyplot as plt
import csv


def add_stock(stock_list):
    option = ""
    while option != "0":
        print("Add a stock:")
        symbol = input("Enter symbol:").upper()
        name = input("Enter company name:")
        shares =float(input("Enter shares: "))
        new_stock = Stock(symbol, name, shares)
        stock_list.append(new_stock)
        option = input("Press enter to add another stock o



# Remove stock and all daily data
def delete_stock(stock_list):
    print("Delete the stock---")
    print("stock list: [ ", end = "")
    for stock in stock_list:
        print(stock.symbol, " ", end = "")
```

```
5 - Show Chart
```

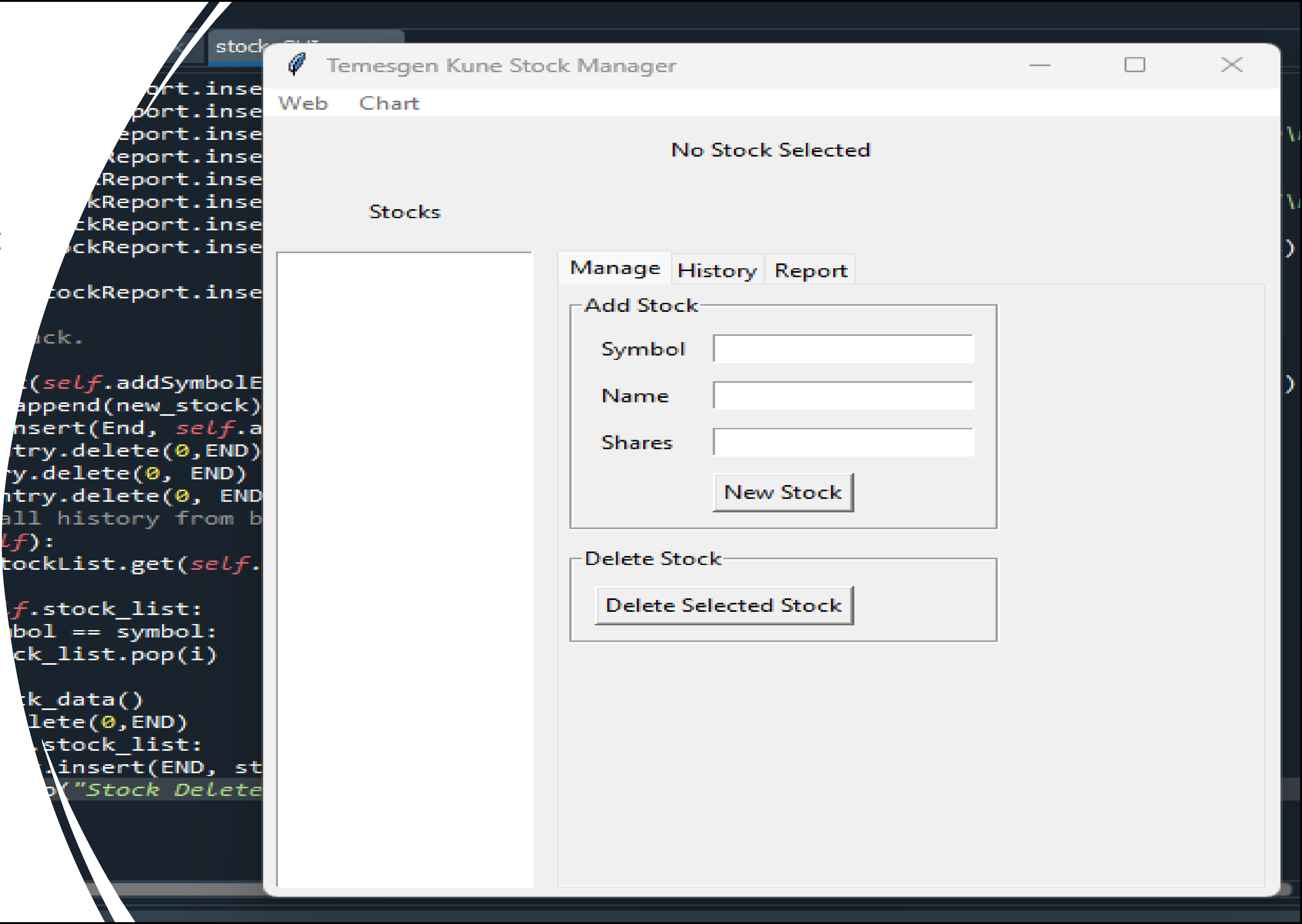
Objectives – Module 7

- Use tkinter to implement a graphical user interface.



Stocks in GUI

- Paste a screen shot of your GUI working

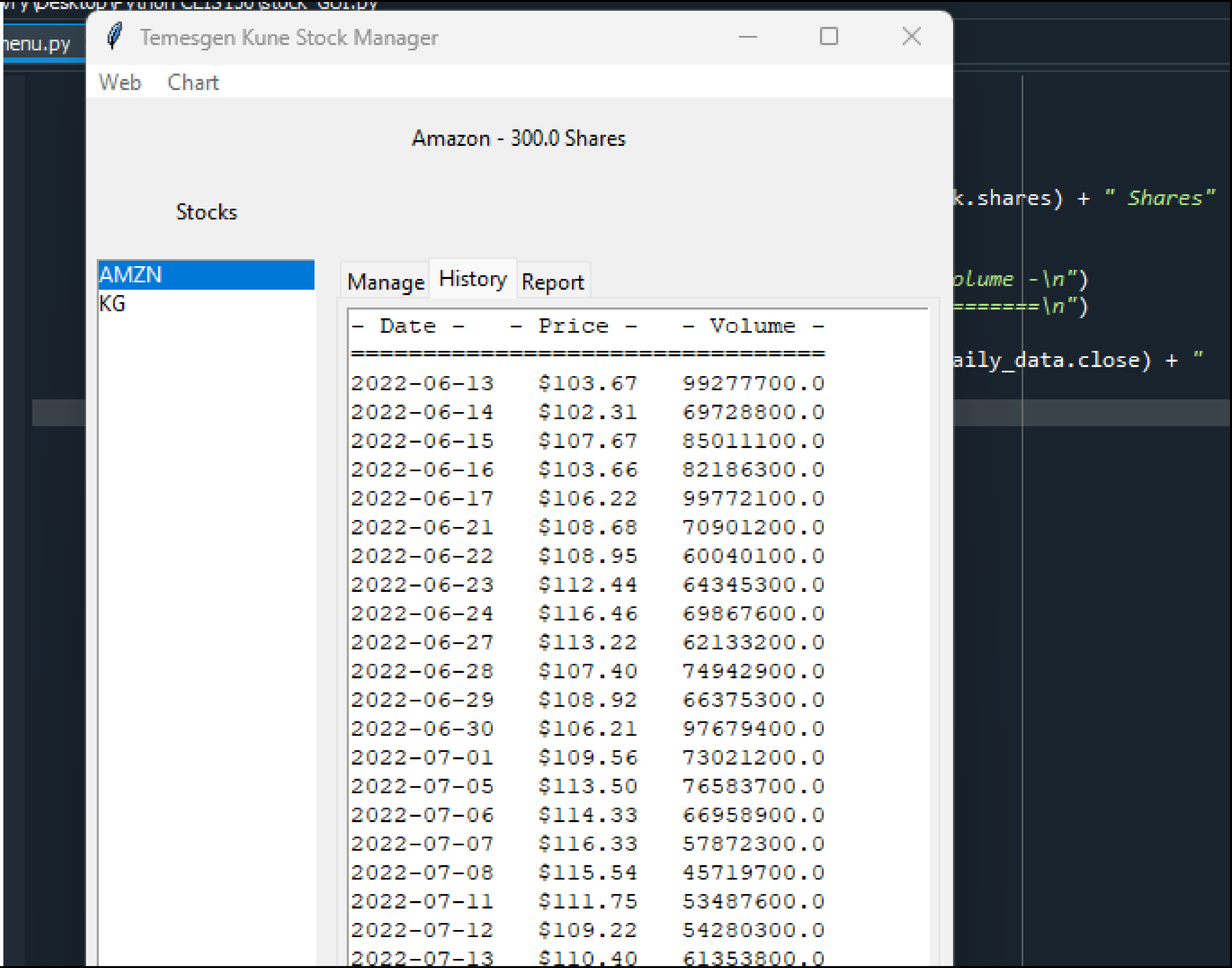


No Stock Selected

```
rt.inse
port.inse
eport.inse
eport.inse
Report.inse
kReport.inse
ckReport.inse
ckReport.inse

ockReport.inse

ack.

(self.addSymbolE
append(new_stock)
nsert(End, self.a
try.delete(0,END)
y.delete(0, END)
ntry.delete(0, END
all history from b
lf):
tockList.get(self.

f.stock_list:
bol == symbol:
ck_list.pop(i)

k_data()
lete(0,END)
.stock_list:
.insert(END, st
o("Stock Delete
```

History Tab

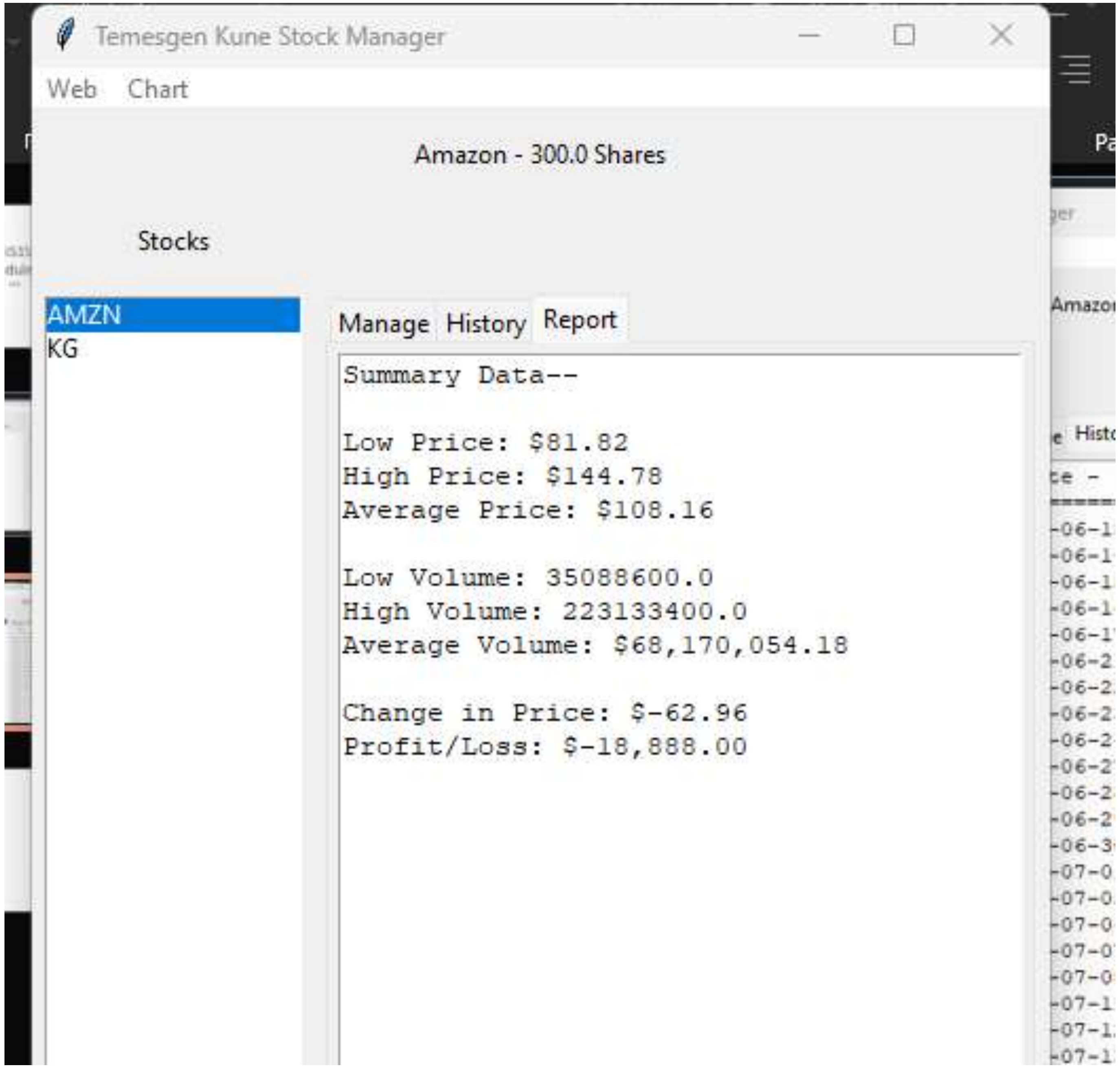
Paste a screen shot of your History tab with import working.



```
- Date -    - Price -    - Volume -
==================================
2022-06-13   $103.67   99277700.0
2022-06-14   $102.31   69728800.0
2022-06-15   $107.67   85011100.0
2022-06-16   $103.66   82186300.0
2022-06-17   $106.22   99772100.0
2022-06-21   $108.68   70901200.0
2022-06-22   $108.95   60040100.0
2022-06-23   $112.44   64345300.0
2022-06-24   $116.46   69867600.0
2022-06-27   $113.22   62133200.0
2022-06-28   $107.40   74942900.0
2022-06-29   $108.92   66375300.0
2022-06-30   $106.21   97679400.0
2022-07-01   $109.56   73021200.0
2022-07-05   $113.50   76583700.0
2022-07-06   $114.33   66958900.0
2022-07-07   $116.33   57872300.0
2022-07-08   $115.54   45719700.0
2022-07-11   $111.75   53487600.0
2022-07-12   $109.22   54280300.0
2022-07-13   $110.40   61353800.0
```

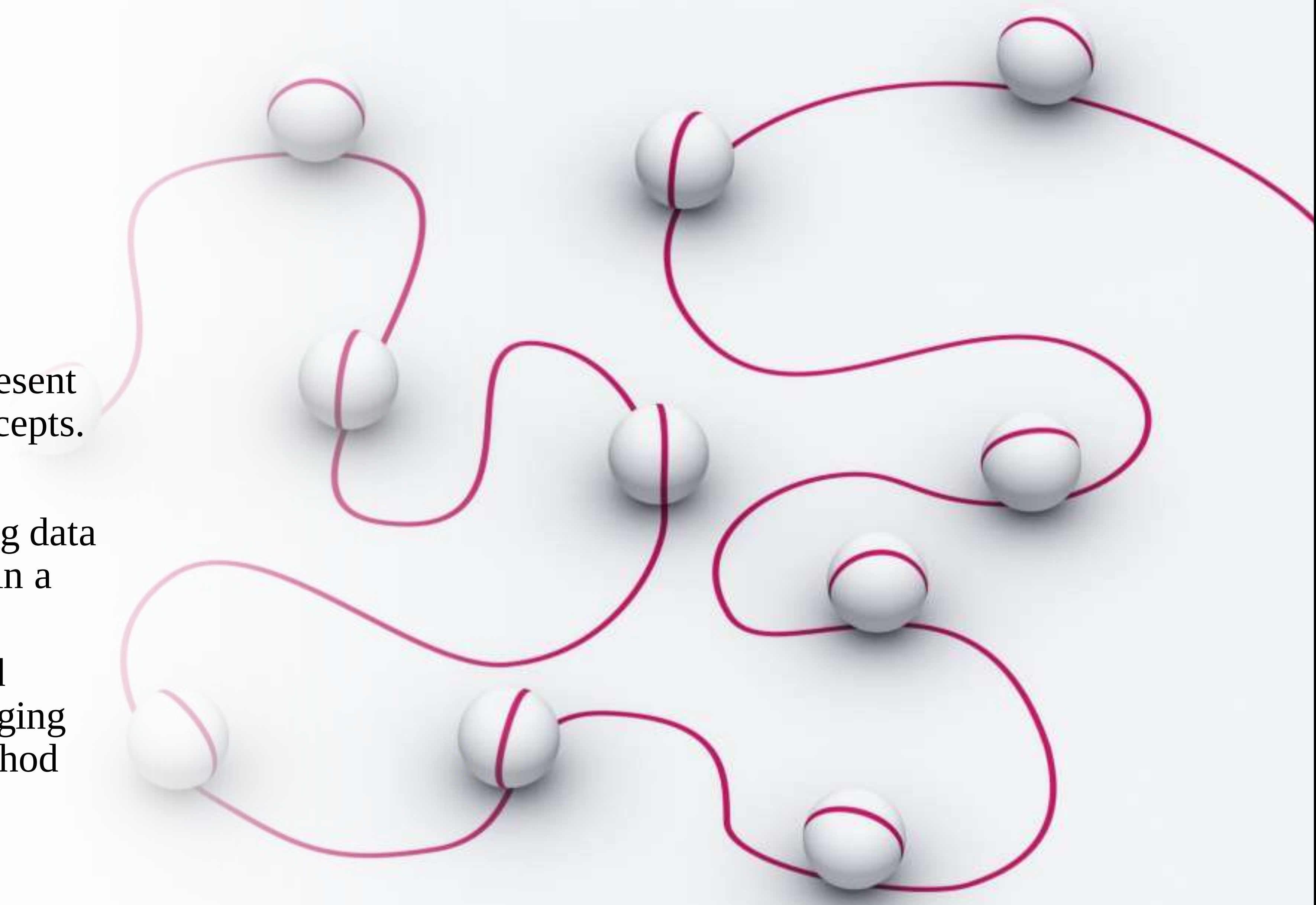
Report Complete

Paste a screen shot of your Report tab



```
Summary Data--

Low Price: $81.82
High Price: $144.78
Average Price: $108.16

Low Volume: 35088600.0
High Volume: 223133400.0
Average Volume: $68,170,054.18

Change in Price: $-62.96
Profit/Loss: $-18,888.00
```

Career Skills Developed Conclusion



- ✓ Creating classes that represent real-world entities or concepts.
- ✓ Encapsulation by bundling data and related methods within a class.
- ✓ Writing more flexible and extensible code by leveraging dynamic binding and method overriding.
- ✓ Analyzing data

Conclusion

- **Object**
  - Object Oriented programing is recommended course for new coders and IT experts.
- **Object**
  - oriented programing is real world problem solver that Every IT personnel
- **Become**
  - one of the most popular programming languages in the world