CODELAB I

ASSESSMENT 2: Utility App

Tutor : Ms. Lavanya Mohan

Programming Fundamentals

Student's Name:	Franz Roland R. Fauni
Student Id	424018
Github Repository Name:	https://github.com/FranzFauni/Franz-Fauni-vending-machine
Github Repository Link:	https://github.com/FranzFauni/Franz-Fauni-vending-machine/blob/main/My%20hopefully%20working%20vending%20machine.py
YouTube Link	https://youtu.be/lUXZPcFhAIM

BRIEF

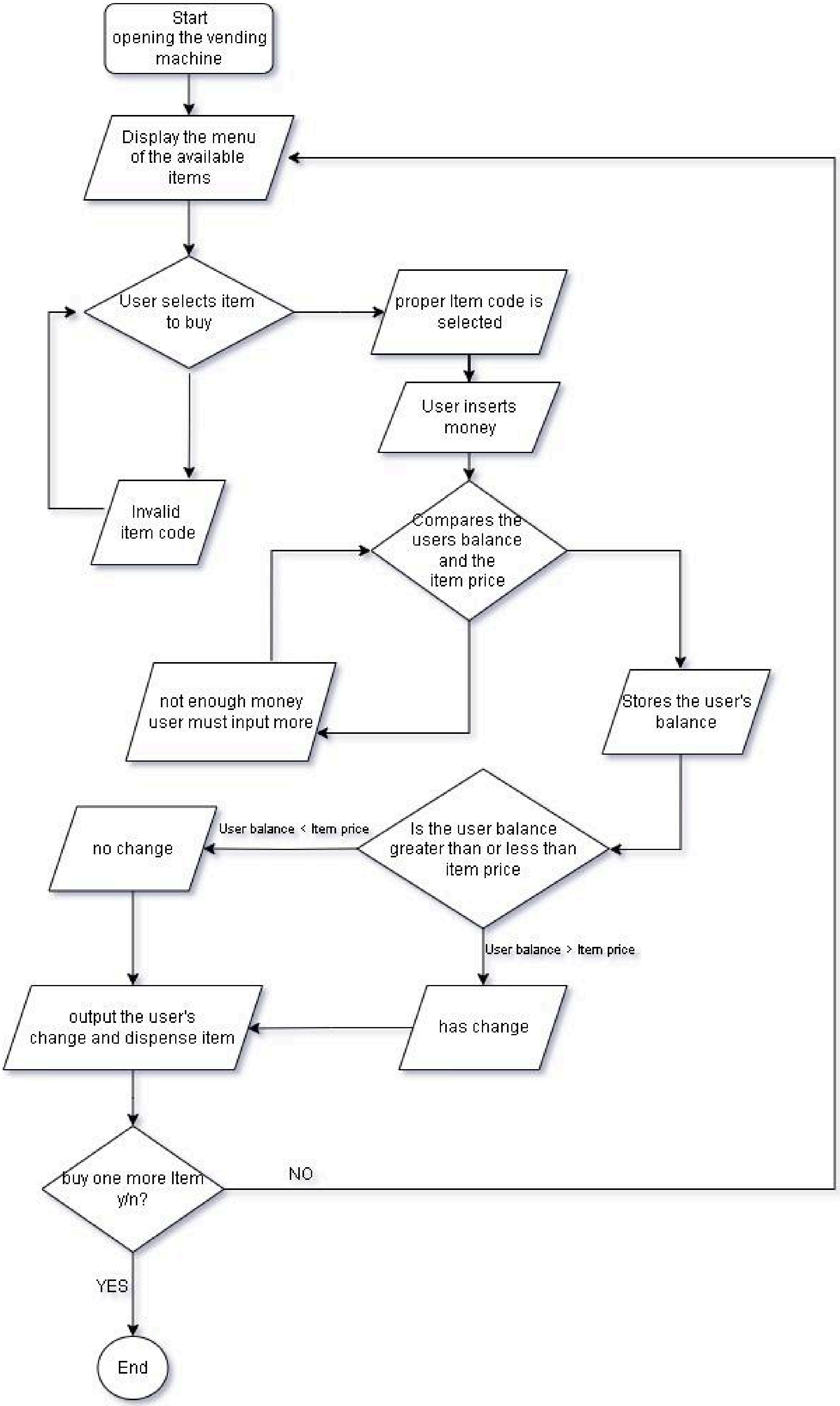
My task is to create a simple working program of a vending machine that does the necessities by using python. This document should show the work that I have done and add my commentary to explain the program that I have created.

Specifications

These are the specifications that I have included into my program. I first of course included the minimum requirements. I am able to present a menu of the available snacks and drinks in the vending machine, allow the user to be able to so select a specific drink or snack of their choosing in the menu, create system that will manage the money of the user so they input the money and get the correct change if needed, tells the user that the item that they bought has been dispensed, give the right amount of change back to the user, and added the comments in the code to show my thought process and what each code is intending to do. I have also added some of the additional features for the vending machine. I added a way so the user is able to buy more items and a way to tell the user an error if they did not input enough money into the vending machine.

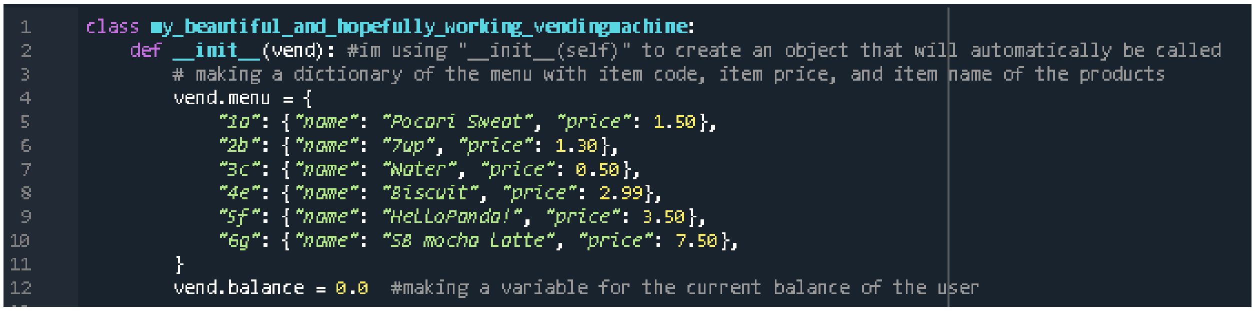
System Flowchart

This is a flowchart I created to help you visualize the path or flow of my vending machine program and how the program solves for the errors or invalid inputs. Essentially the main idea of the program.


Start
opening the vending machine
Display the menu of the available items
User selects item to buy
proper Item code is selected
Invalid item code
User inserts money
Compares the users balance and the item price
not enough money user must input more
Stores the user's balance
Is the user balance greater than or less than item price
User balance < Item price
no change
User balance > Item price
has change
output the user's change and dispense item
buy one more Item y/n?
NO
YES
End

Technical Description

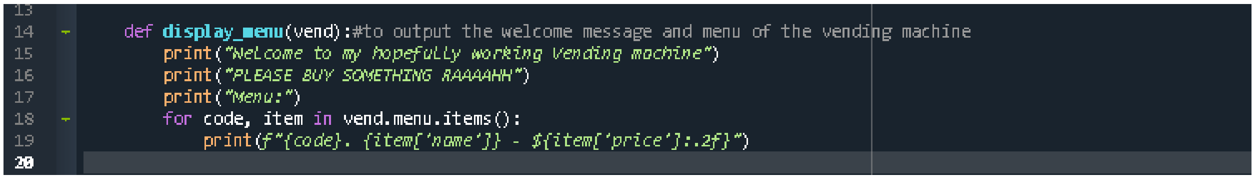
Before I explain the blocks of code I did for this program, I will tell you the methodology in creating the program from scratch. The approach I decided to do for this vending machine program is quite simple. First make a class on which I will be putting every function for the program, second to create every action or specification in on its own function to make it clean, and third call the class to run the program.



```python
class my_beautiful_and_hopefully_working_vendingmachine:
    def __init__(vend): #im using "__init__(self)" to create an object that will automatically be called
        # making a dictionary of the menu with item code, item price, and item name of the products
        vend.menu = {
            "1a": {"name": "Pocari Sweat", "price": 1.50},
            "2b": {"name": "7up", "price": 1.30},
            "3c": {"name": "Water", "price": 0.50},
            "4e": {"name": "Biscuit", "price": 2.99},
            "5f": {"name": "HelloPanda!", "price": 3.50},
            "6g": {"name": "S8 mocha Latte", "price": 7.50},
        }
        vend.balance = 0.0  #making a variable for the current balance of the user
```

figure 1

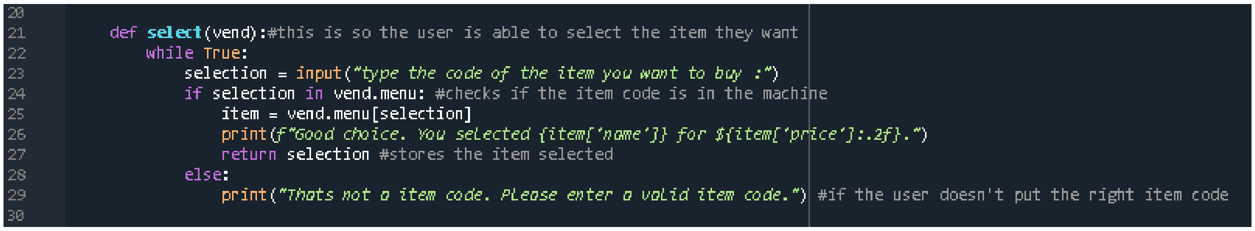
As you can see on figure 1 this is the beginning of the class. One reason I decided to put the main core of the program in a class is because I can use the "__init__" method. What this essentially does is initiates the attributes that will automatically be referred to every function that I created afterwards. There are two things that will be automatically referred to by the other functions, the dictionary of the menu items, and the user balance. So the dictionary consists of the items of the vending machine. Every item has its own Id number, name, and the price, this will be the menu that the user will see. And lastly this initiating function will have a variable called "vend. balance" that will be storing the user balance.



```python
    def display_menu(vend):#to output the welcome message and menu of the vending machine
        print("Welcome to my hopefully working Vending machine")
        print("PLEASE BUY SOMETHING RAAAAHH")
        print("Menu:")
        for code, item in vend.menu.items():
            print(f"{code}. {item['name']} - ${item['price']:.2f}")
```

figure 2

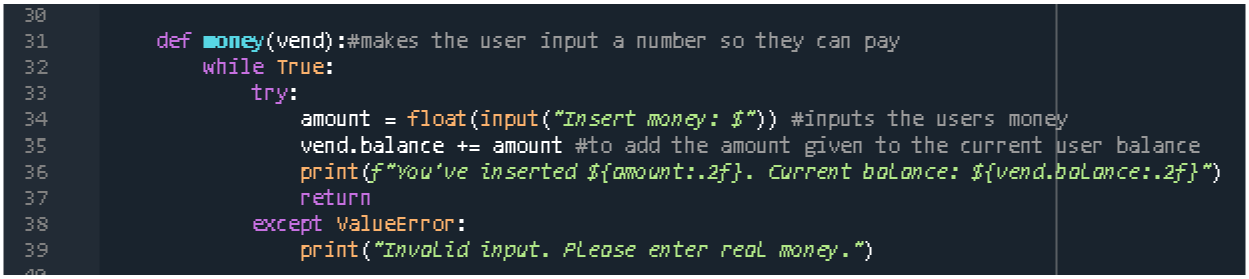
In figure number 2 this is the next function I created. The main purpose of this function is to display the welcome message to the user and to display the menu of items that are available in the vending machine. It only uses simple print messages and only calls the dictionary to be printed in order. And with that the user can now see the menu and choose what item that they would like.



```python
def select(vend):#this is so the user is able to select the item they want
    while True:
        selection = input("type the code of the item you want to buy :")
        if selection in vend.menu: #checks if the item code is in the machine
            item = vend.menu[selection]
            print(f"Good choice. You selected {item['name']} for ${item['price']:.2f}.")
            return selection #stores the item selected
        else:
            print("Thats not a item code. Please enter a valid item code.") #if the user doesn't put the right item code
```

Figure 3

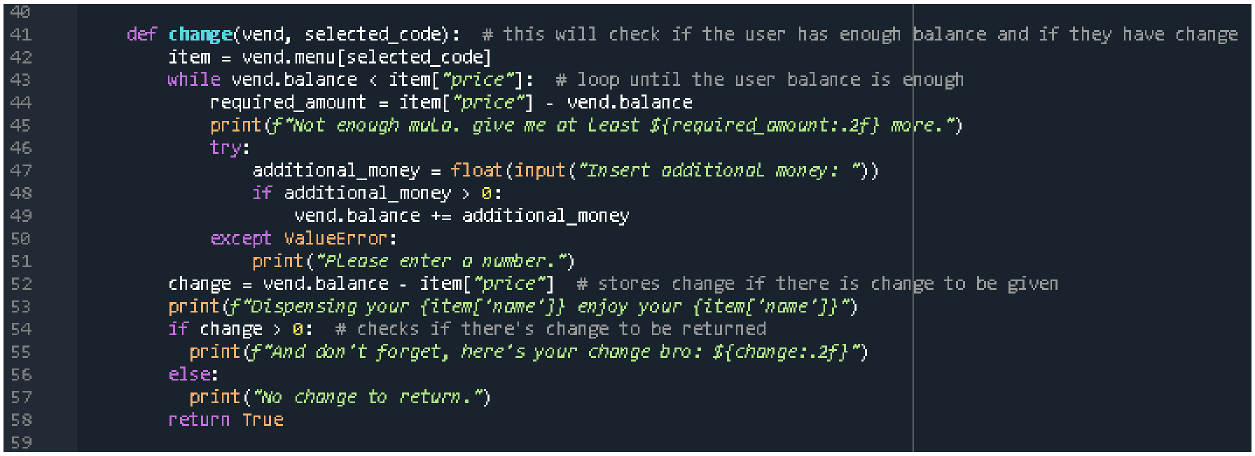
For the main purpose of this function it will allow the user to be able to select any item they like in the menu of the vending machine. Using an input line it will store the user's selection and then will go through an if else statement. If the user selects an item successfully by entering the proper item code of the item they want it will be stored on a different variable called “item” and will print out a message to the user what item they have selected and its price. and if the user does not enter the proper item code of the item they wanted then the else will trigger an invalid message. It will print out a message saying the user did not enter a proper code and will ask to repeat the item code they intended to put for the item they wanted to select.



```python
def money(vend):#makes the user input a number so they can pay
    while True:
        try:
            amount = float(input("Insert money: $")) #inputs the users money
            vend.balance += amount #to add the amount given to the current user balance
            print(f"You've inserted ${amount:.2f}. Current balance: ${vend.balance:.2f}")
            return
        except ValueError:
            print("Invalid input. Please enter real money.")
```

Figure 4

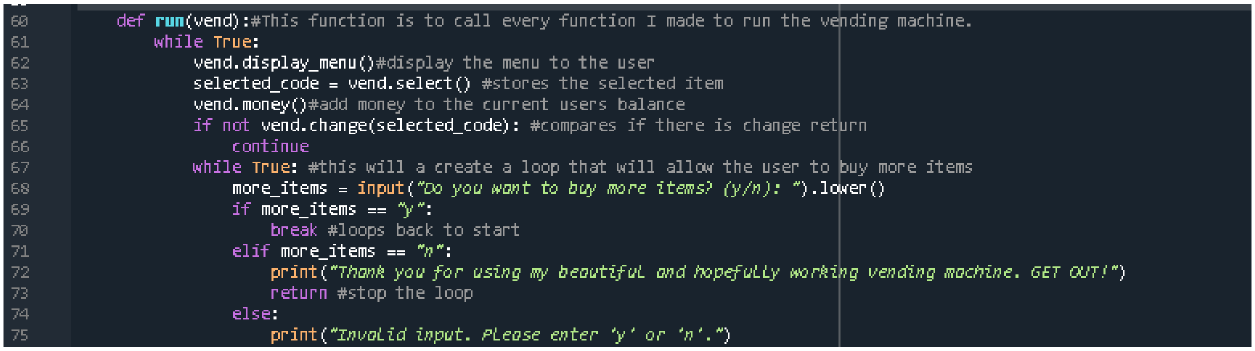
For this function it will allow the user to input money so they can buy the item they have selected. So first the user will be prompted to input money and it will be stored in the “amount” variable, next the amount will be added to the “vend.balance” variable from the start so it will be saved. And then the user will be given a message saying the amount of money they have added to the vending machine and the overall current balance that they have inside the vending machine. And if the user did not put a proper numerical value when they are prompted to do so, It will trigger an error message saying that they have put an invalid and will be asked again to give a proper numerical value.



```
    def change(vend, selected_code):  # this will check if the user has enough balance and if they have change
        item = vend.menu[selected_code]
        while vend.balance < item[“price”]:  # loop until the user balance is enough
            required_amount = item[“price”] - vend.balance
            print(f“Not enough mula. give me at least ${required_amount:.2f} more.”)
            try:
                additional_money = float(input(“Insert additional money: ”))
                if additional_money > 0:
                    vend.balance += additional_money
            except ValueError:
                print(“Please enter a number.”)
        change = vend.balance - item[“price”]  # stores change if there is change to be given
        print(f“Dispensing your {item[‘name’]} enjoy your {item[‘name’]}”)
        if change > 0:  # checks if there's change to be returned
          print(f“And don’t forget, here’s your change bro: ${change:.2f}”)
        else:
          print(“No change to return.”)
        return True
```

figure 5

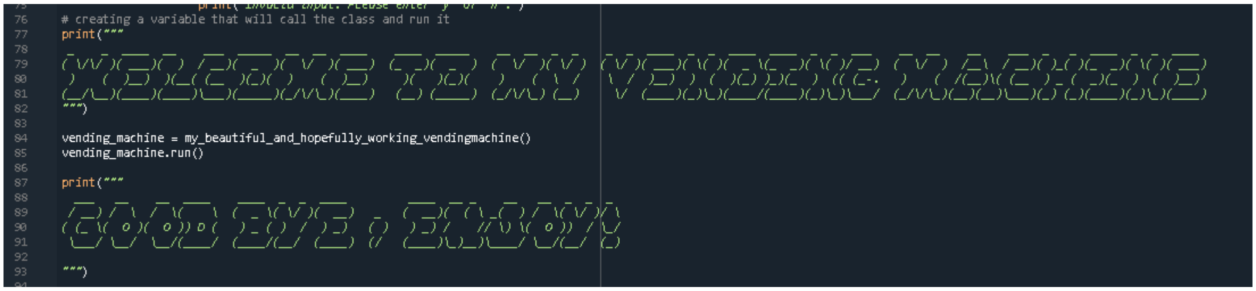
Unlike the past functions I have shown, This function has two purposes. First is to check if the user has given enough money in their balance to buy the item they have selected and second to see if the user will receive change. The code will firstly compare the user balance and the item price of the item they have chosen. If the user balance is less than the item price it will subtract the item price to the user balance to know how much the user needs to give and will print out a message that will tell the user how much they need to give in order to pay. It will then allow the user to pay the amount they need and store that entered amount again to the user balance of the vending machine. When the user balance is sufficient enough, the code will now subtract again the user balance to the item price of the item selected and store it in a variable named “change”. If the change is greater than 0 then the code will print out the change and if the change is equal to 0 then it will message the user that there is no change to be given. And lastly of course it will message the user that the item selected has been dispensed.



```
    def run(vend):#This function is to call every function I made to run the vending machine.
        while True:
            vend.display_menu()#display the menu to the user
            selected_code = vend.select() #stores the selected item
            vend.money()#add money to the current users balance
            if not vend.change(selected_code): #compares if there is change return
                continue
            while True: #this will a create a loop that will allow the user to buy more items
                more_items = input(“Do you want to buy more items? (y/n): ”).lower()
                if more_items == “y”:
                    break #loops back to start
                elif more_items == “n”:
                    print(“Thank you for using my beautiful and hopefully working vending machine. GET OUT!”)
                    return #stop the loop
                else:
                    print(“Invalid input. Please enter ‘y’ or ‘n’.”)
```

Figure 6

And then lastly this last function will call every other function so it will run properly. First the menu displayed for the user, the item selection, then allows the user to enter money so they can pay the selected item, and then compares the user balance to the item price to see if the user gave enough money and if they need change it the user balance is sufficient. And after calling every function it will then ask if the user wants to buy more items. If yes then it will go back to the menu display, and if not it will end there and give the goodbye message.



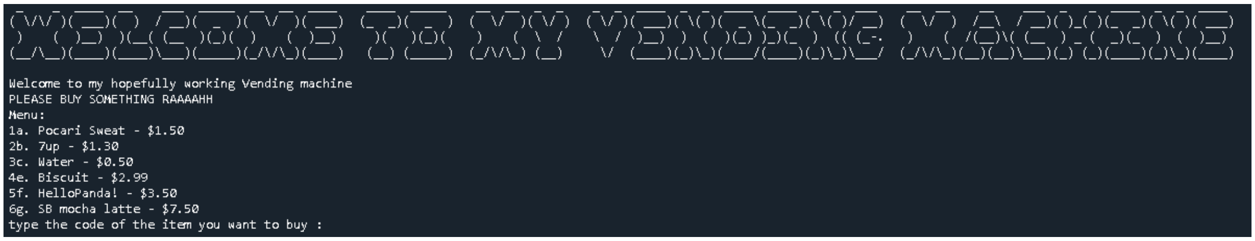
```
# creating a variable that will call the class and run it
print("""

""")

vending_machine = my_beautiful_and_hopefully_working_vendingmachine()
vending_machine.run()

print("""

""")
```

figure 7

And finally after finishing every function I need in the class I then created a welcome design for the user in the beginning and then called the class so the program can run. And when the program is done it will also print out a last goodbye message to the user.

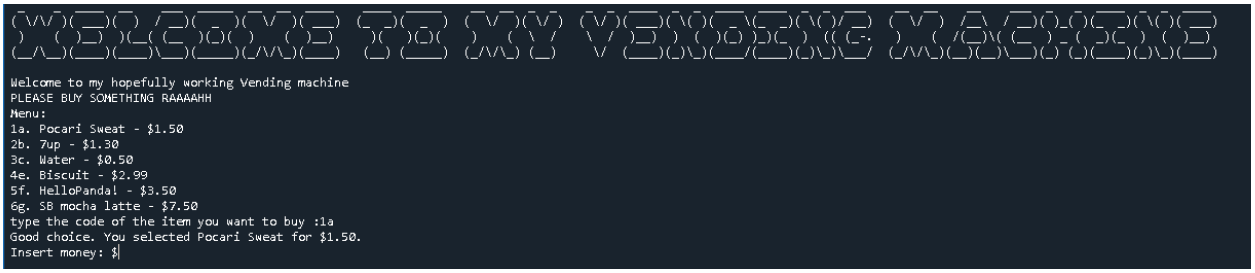
Output

Opening menu display:



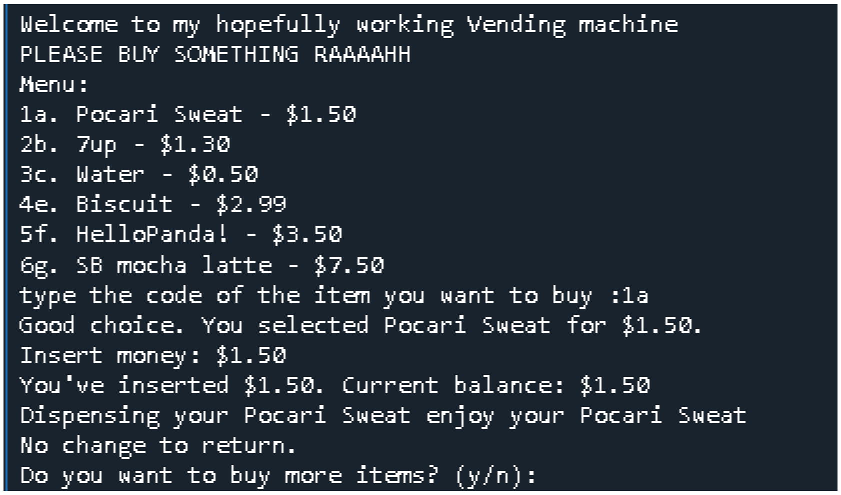
```
Welcome to my hopefully working Vending machine
PLEASE BUY SOMETHING RAAAAHH
Menu:
1a. Pocari Sweat - $1.50
2b. 7up - $1.30
3c. Water - $0.50
4e. Biscuit - $2.99
5f. HelloPanda! - $3.50
6g. SB mocha latte - $7.50
type the code of the item you want to buy :
```

After user has selected an item:



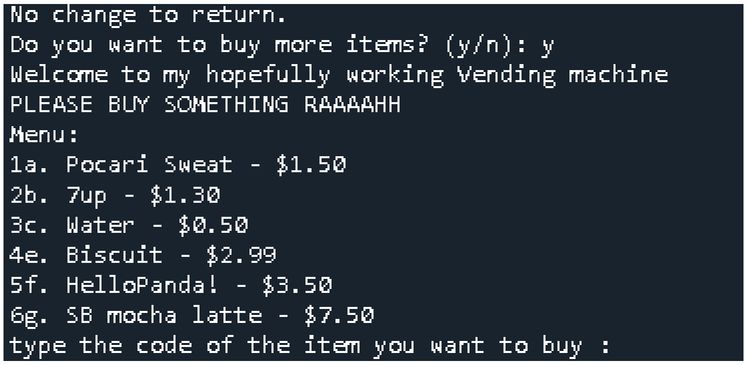
```
Welcome to my hopefully working Vending machine
PLEASE BUY SOMETHING RAAAAHH
Menu:
1a. Pocari Sweat - $1.50
2b. 7up - $1.30
3c. Water - $0.50
4e. Biscuit - $2.99
5f. HelloPanda! - $3.50
6g. SB mocha latte - $7.50
type the code of the item you want to buy :1a
Good choice. You selected Pocari Sweat for $1.50.
Insert money: $
```

When user has given sufficient money for balance:



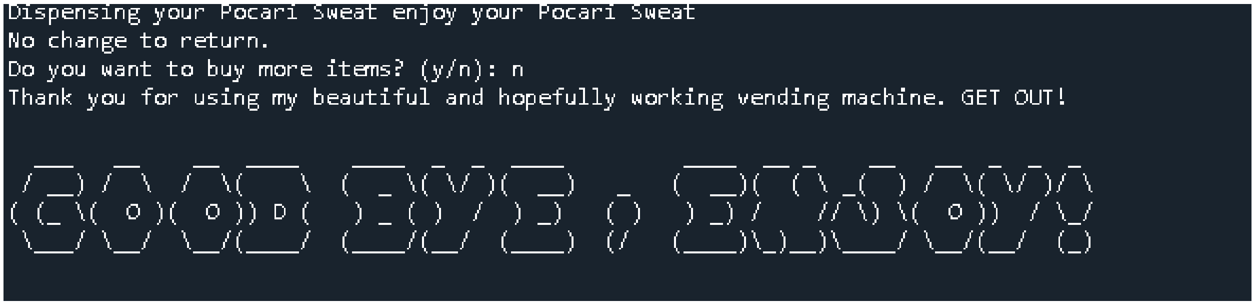
```
Welcome to my hopefully working Vending machine
PLEASE BUY SOMETHING RAAAAHH
Menu:
1a. Pocari Sweat - $1.50
2b. 7up - $1.30
3c. Water - $0.50
4e. Biscuit - $2.99
5f. HelloPanda! - $3.50
6g. SB mocha latte - $7.50
type the code of the item you want to buy :1a
Good choice. You selected Pocari Sweat for $1.50.
Insert money: $1.50
You've inserted $1.50. Current balance: $1.50
Dispensing your Pocari Sweat enjoy your Pocari Sweat
No change to return.
Do you want to buy more items? (y/n):
```

If user wants to buy more item:



```
No change to return.
Do you want to buy more items? (y/n): y
Welcome to my hopefully working Vending machine
PLEASE BUY SOMETHING RAAAAHH
Menu:
1a. Pocari Sweat - $1.50
2b. 7up - $1.30
3c. Water - $0.50
4e. Biscuit - $2.99
5f. HelloPanda! - $3.50
6g. SB mocha latte - $7.50
type the code of the item you want to buy :
```

Last goodbye message when user say no to more items:



```
Dispensing your Pocari Sweat enjoy your Pocari Sweat
No change to return.
Do you want to buy more items? (y/n): n
Thank you for using my beautiful and hopefully working vending machine. GET OUT!


  ___   __    __  ____     ____  _  _  ____       ____  __ _     __   __  _  _  _
 / __) /  \  /  \(    \   (  _ \( \/ )(  __)  _   (  __)(  ( \ _(  ) /  \( \/ )/ \
( (_ \(  O )(  O )) D (    ) _ ( )  /  ) _)  ( )   ) _) /    // \) \(  O ))  / \_/
 \___/ \__/  \__/(____/   (____/(__/  (____) (/   (____)\_)__)\____/ \__/(__/  (_)
```

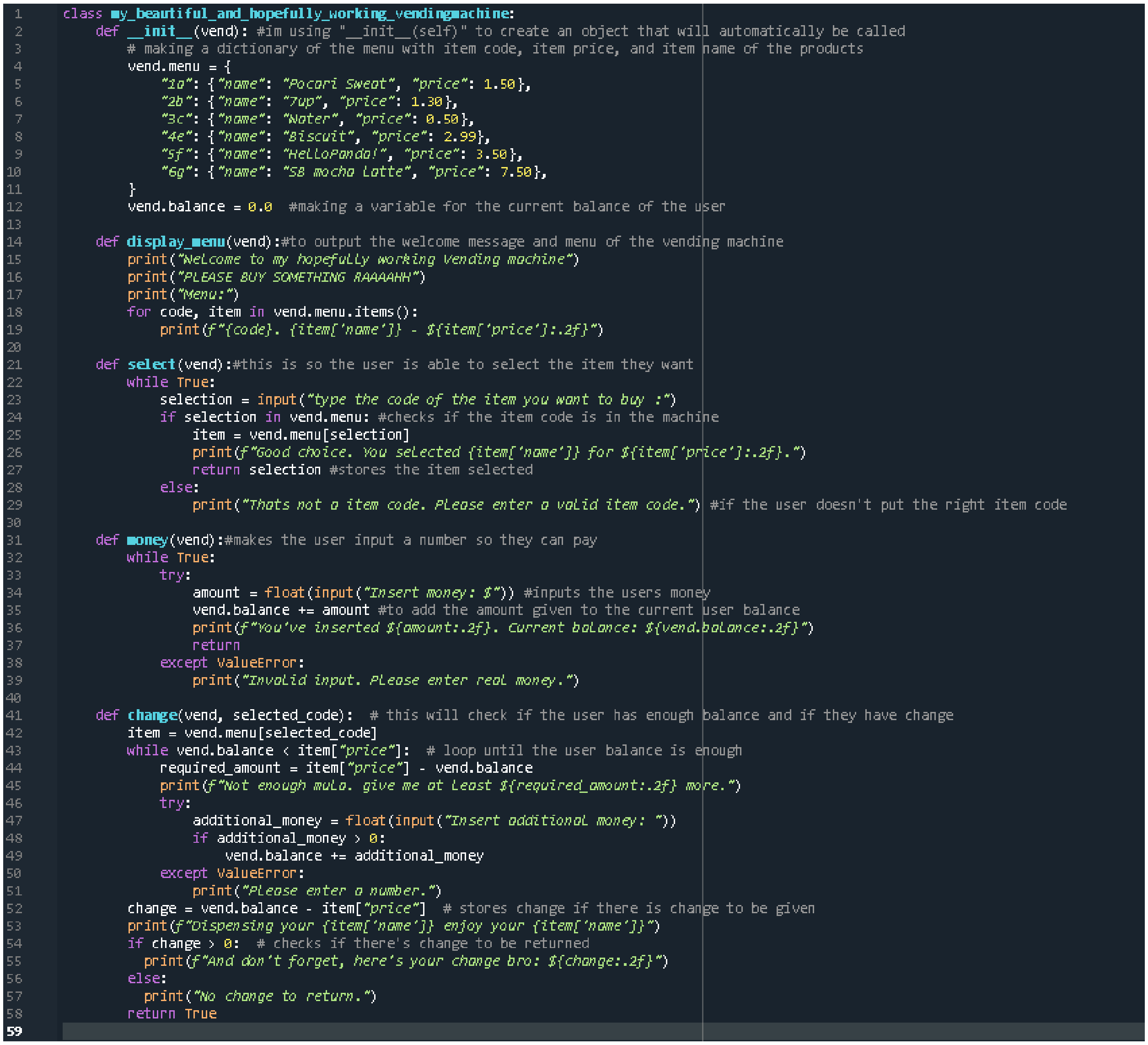
Critical Reflection

Creating the vending machine program has definitely been a fun challenge. Before this activity the only experience was the small coding adventures in intro to programming class. Besides in class I have not really touched python. But with this specific activity I had to relearn a few things and even learn new things about python. That process of getting to know and creating this program piece by piece really brought appreciation in coding. After this activity I can really see myself trying to find a new project to do to enhance my knowledge with python and maybe even try and learn a different coding language.

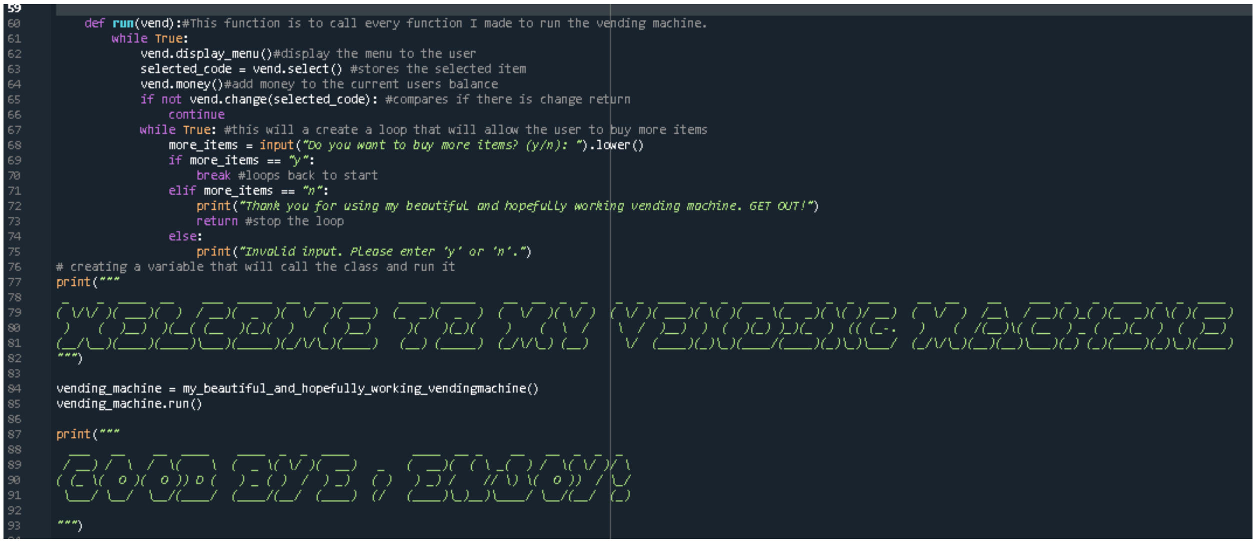
With the vending machine program I have created, I can see that there is a lot more room to improve and maybe even some bravery to add more features. I was not too confident I was able to apply to the program. But overall I am really satisfied and happy with the program I have created. My goal was to create a simple and complete vending machine and I believe I have accomplished that goal.

After this assessment I can confidently say that I am now comfortable with using the python coding language. And that I still have retained the past lessons we had and even learned new things during this. Although I was not excited in this activity I slowly found the charm in finding the solution to the problems I will eventually encounter. And I found excitement when the code I have been working on finally works.

Appendix



```python
class my_beautiful_and_hopefully_working_vendingmachine:
    def __init__(vend): #im using "__init__(self)" to create an object that will automatically be called
        # making a dictionary of the menu with item code, item price, and item name of the products
        vend.menu = {
            "1a": {"name": "Pocari Sweat", "price": 1.50},
            "2b": {"name": "7up", "price": 1.30},
            "3c": {"name": "Water", "price": 0.50},
            "4e": {"name": "Biscuit", "price": 2.99},
            "5f": {"name": "HelloPanda!", "price": 3.50},
            "6g": {"name": "58 mocha Latte", "price": 7.50},
        }
        vend.balance = 0.0  #making a variable for the current balance of the user

    def display_menu(vend):#to output the welcome message and menu of the vending machine
        print("Welcome to my hopefully working vending machine")
        print("PLEASE BUY SOMETHING RAAAAHH")
        print("Menu:")
        for code, item in vend.menu.items():
            print(f"{code}. {item['name']} - ${item['price']:.2f}")

    def select(vend):#this is so the user is able to select the item they want
        while True:
            selection = input("type the code of the item you want to buy :")
            if selection in vend.menu: #checks if the item code is in the machine
                item = vend.menu[selection]
                print(f"Good choice. You selected {item['name']} for ${item['price']:.2f}.")
                return selection #stores the item selected
            else:
                print("Thats not a item code. Please enter a valid item code.") #if the user doesn't put the right item code

    def money(vend):#makes the user input a number so they can pay
        while True:
            try:
                amount = float(input("Insert money: $")) #inputs the users money
                vend.balance += amount #to add the amount given to the current user balance
                print(f"You've inserted ${amount:.2f}. Current balance: ${vend.balance:.2f}")
                return
            except ValueError:
                print("Invalid input. Please enter real money.")

    def change(vend, selected_code):  # this will check if the user has enough balance and if they have change
        item = vend.menu[selected_code]
        while vend.balance < item["price"]:  # loop until the user balance is enough
            required_amount = item["price"] - vend.balance
            print(f"Not enough mula. give me at least ${required_amount:.2f} more.")
            try:
                additional_money = float(input("Insert additional money: "))
                if additional_money > 0:
                    vend.balance += additional_money
            except ValueError:
                print("Please enter a number.")
        change = vend.balance - item["price"]  # stores change if there is change to be given
        print(f"Dispensing your {item['name']} enjoy your {item['name']}")
        if change > 0:  # checks if there's change to be returned
          print(f"And don't forget, here's your change bro: ${change:.2f}")
        else:
          print("No change to return.")
        return True

```



```python

    def run(vend):#This function is to call every function I made to run the vending machine.
        while True:
            vend.display_menu()#display the menu to the user
            selected_code = vend.select() #stores the selected item
            vend.money()#add money to the current users balance
            if not vend.change(selected_code): #compares if there is change return
                continue
            while True: #this will a create a loop that will allow the user to buy more items
                more_items = input("Do you want to buy more items? (y/n): ").lower()
                if more_items == "y":
                    break #loops back to start
                elif more_items == "n":
                    print("Thank you for using my beautiful and hopefully working vending machine. GET OUT!")
                    return #stop the loop
                else:
                    print("Invalid input. Please enter 'y' or 'n'.")
# creating a variable that will call the class and run it
print("""
 _    _  ____  __    ___  _____  __  __  ____    ____  _____    __  __  _  _    _  _  ____  _  _  ____  ____  _  _  ___    __  __    __    ___  _   _  ____  _  _  ____ 
( \/\/ )( ___)(  )  / __)(  _  )(  \/  )( ___)  (_  _)(  _  )  (  \/  )( \/ )  ( \/ )( ___)( \( )(  _ \(_  _)( \( )/ __)  (  \/  )  /__\  / __)( )_( )(_  _)( \( )( ___)
 )    (  )__)  )(__( (__  )(_)(  )    (  )__)    )(   )(_)(    )    (  \  /    \  /  )__)  )  (  )(_) )_)(_  )  (( (_-.   )    (  /(__)\( (__  ) _ (  _)(_  )  (  )__) 
(__/\__)(____)(____)\___)(_____)(_/\/\_)(____)  (__) (_____)  (_/\/\_) (__)    \/  (____)(_)\_)(____/(____)(_)\_)\___/  (_/\/\_)(__)(__)\___)(_) (_)(____)(_)\_)(____)
""")

vending_machine = my_beautiful_and_hopefully_working_vendingmachine()
vending_machine.run()

print("""
  ___   __    __  ____    ____  _  _  ____      ____  __ _    __   __  _  _  _ 
 / __) /  \  /  \(    \  (  _ \( \/ )(  __)  _  (  __)(  ( \ _(  ) /  \( \/ )/ \
( (_ \(  O )(  O )) D (   ) _ ( )  /  ) _)  ( )  ) _) /    // \) \(  O ))  / \_/
 \___/ \__/  \__/(____/  (____/(__/  (____) (/  (____)\_)__)\____/ \__/(__/  (_)

""")
```

Reference List

PythonDex. (n.d.). *Vending machine Python program*. Available at: https://pythondex.com/vending-machine-python-program

freeCodeCamp.org. (2021). *Python vending machine project tutorial*. Available at: https://www.youtube.com/watch?v=WAQMcsTKzwM

OpenAI. (2025). *ChatGPT*. Available at: https://chat.openai.com/