**Pseudocode:**

1. ItemToPurchase Class:

- Attributes:

- `item\_name`: string, default "none"

- `item\_price`: float, default 0.0

- `item\_quantity`: int, default 0

- `item\_description`: string, default "none"

- Methods:

- `\_\_init\_\_()`: Initialize the item with default values.

- `print\_item\_cost()`: Print the item's name, quantity, price, and total cost.

- `print\_item\_description()`: Print the item's name and description.

2. ShoppingCart Class:

- Attributes:

- `customer\_name`: string, default "none"

- `current\_date`: string, default "January 1, 2020"

- `cart\_items`: list of `ItemToPurchase` objects

- Methods:

- `\_\_init\_\_()`: Initialize the shopping cart with default values.

- `add\_item(item)`: Add an `ItemToPurchase` object to the cart.

- `remove\_item(item\_name)`: Remove an item by its name from the cart.

- `modify\_item(item)`: Modify the price, quantity, or description of an item in the cart.

- `get\_num\_items\_in\_cart()`: Return the total quantity of items in the cart.

- `get\_cost\_of\_cart()`: Return the total cost of the items in the cart.

- `print\_total()`: Print the total number of items and the total cost.

- `print\_descriptions()`: Print the description of all items in the cart.

3. print\_menu(cart):

- Display menu options:

- `a`: Add item to cart

- `r`: Remove item from cart

- `c`: Change item quantity

- `i`: Output items' descriptions

- `o`: Output shopping cart total

- `q`: Quit

- Loop:

- Prompt the user to choose an option.

- Call the respective method based on the option chosen.

- Quit when the user chooses `q`.

4. main() Function:

- Prompt the user for their name and today's date.

- Create a `ShoppingCart` object using the name and date.

- Call `print\_menu(cart)` to start the program and allow user interaction.

**Source code:**

# Define the ItemToPurchase class

class ItemToPurchase:

def \_\_init\_\_(self, name="none", price=0.0, quantity=0, description="none"):

self.item\_name = name

self.item\_price = price

self.item\_quantity = quantity

self.item\_description = description

# Method to print item cost

def print\_item\_cost(self):

total\_cost = self.item\_price \* self.item\_quantity

print(f'{self.item\_name} {self.item\_quantity} @ ${self.item\_price} = ${total\_cost}')

# Method to print item description

def print\_item\_description(self):

print(f"{self.item\_name}: {self.item\_description}")

# Define the ShoppingCart class

class ShoppingCart:

def \_\_init\_\_(self, customer\_name="none", current\_date="January 1, 2020"):

self.customer\_name = customer\_name

self.current\_date = current\_date

self.cart\_items = []

# Adds an item to the cart

def add\_item(self, item\_to\_purchase):

self.cart\_items.append(item\_to\_purchase)

# Removes an item from the cart by name

def remove\_item(self, item\_name):

item\_found = False

for item in self.cart\_items:

if item.item\_name == item\_name:

self.cart\_items.remove(item)

item\_found = True

break

if not item\_found:

print("Item not found in cart. Nothing removed.")

# Modifies an item’s description, price, or quantity

def modify\_item(self, modified\_item):

item\_found = False

for item in self.cart\_items:

if item.item\_name == modified\_item.item\_name:

if modified\_item.item\_description != "none":

item.item\_description = modified\_item.item\_description

if modified\_item.item\_price != 0:

item.item\_price = modified\_item.item\_price

if modified\_item.item\_quantity != 0:

item.item\_quantity = modified\_item.item\_quantity

item\_found = True

break

if not item\_found:

print("Item not found in cart. Nothing modified.")

# Returns total number of items in the cart

def get\_num\_items\_in\_cart(self):

return sum(item.item\_quantity for item in self.cart\_items)

# Calculates and returns the total cost of the cart

def get\_cost\_of\_cart(self):

total\_cost = 0

for item in self.cart\_items:

total\_cost += item.item\_price \* item.item\_quantity

return total\_cost

# Prints the total cost of the shopping cart

def print\_total(self):

if len(self.cart\_items) == 0:

print("SHOPPING CART IS EMPTY")

else:

print(f"{self.customer\_name}'s Shopping Cart - {self.current\_date}")

print(f"Number of Items: {self.get\_num\_items\_in\_cart()}")

for item in self.cart\_items:

item.print\_item\_cost()

print(f"Total: ${self.get\_cost\_of\_cart()}")

# Prints descriptions of each item in the cart

def print\_descriptions(self):

if len(self.cart\_items) == 0:

print("SHOPPING CART IS EMPTY")

else:

print(f"{self.customer\_name}'s Shopping Cart - {self.current\_date}")

print("Item Descriptions")

for item in self.cart\_items:

item.print\_item\_description()

# Function to display the menu and process user input

def print\_menu(cart):

while True:

print("\nMENU")

print("a - Add item to cart")

print("r - Remove item from cart")

print("c - Change item quantity")

print("i - Output items' descriptions")

print("o - Output shopping cart")

print("q - Quit")

choice = input("Choose an option: ").lower()

if choice == 'a':

name = input("Enter item name: ")

description = input("Enter item description: ")

price = float(input("Enter item price:$ "))

quantity = int(input("Enter item quantity: "))

item = ItemToPurchase(name, price, quantity, description)

cart.add\_item(item)

elif choice == 'r':

name = input("Enter the name of the item to remove: ")

cart.remove\_item(name)

elif choice == 'c':

name = input("Enter the name of the item to modify: ")

description = input("Enter the new description (or 'none' to leave unchanged): ")

price = float(input("Enter the new price (or 0 to leave unchanged): "))

quantity = int(input("Enter the new quantity (or 0 to leave unchanged): "))

modified\_item = ItemToPurchase(name, price, quantity, description)

cart.modify\_item(modified\_item)

elif choice == 'i':

cart.print\_descriptions()

elif choice == 'o':

cart.print\_total()

elif choice == 'q':

break

else:

print("Invalid option, try again.")

# Main function to start the program

def main():

customer\_name = input("Enter customer's name: ")

current\_date = input("Enter today's date: ")

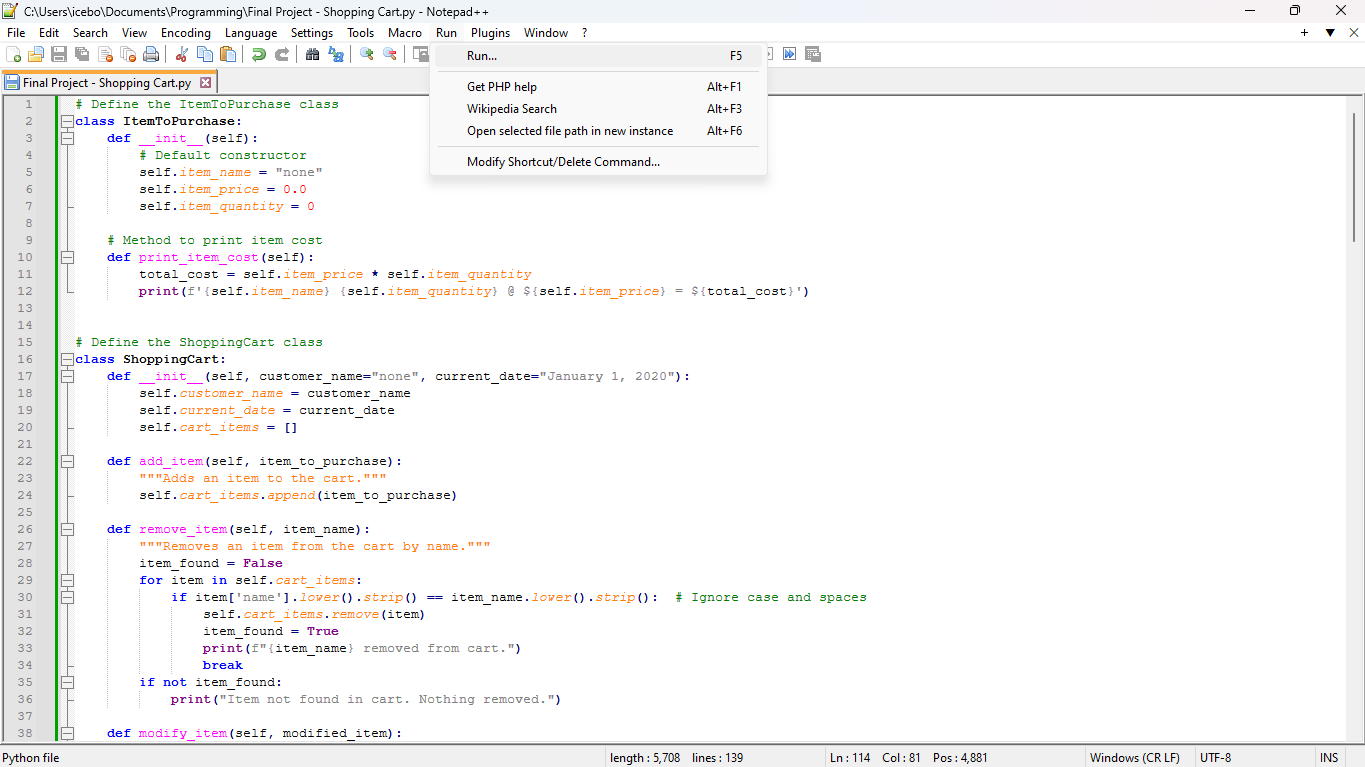
cart = ShoppingCart(customer\_name, current\_date)

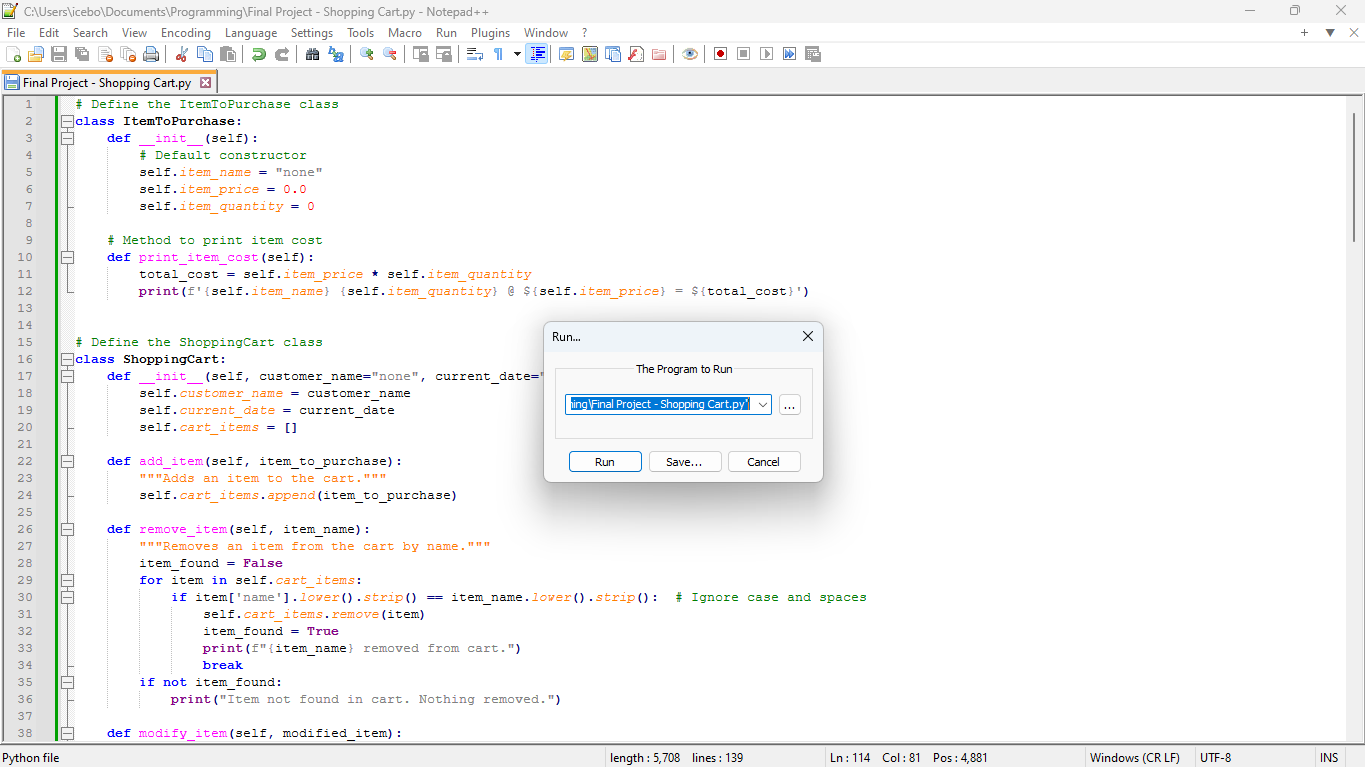
print\_menu(cart)

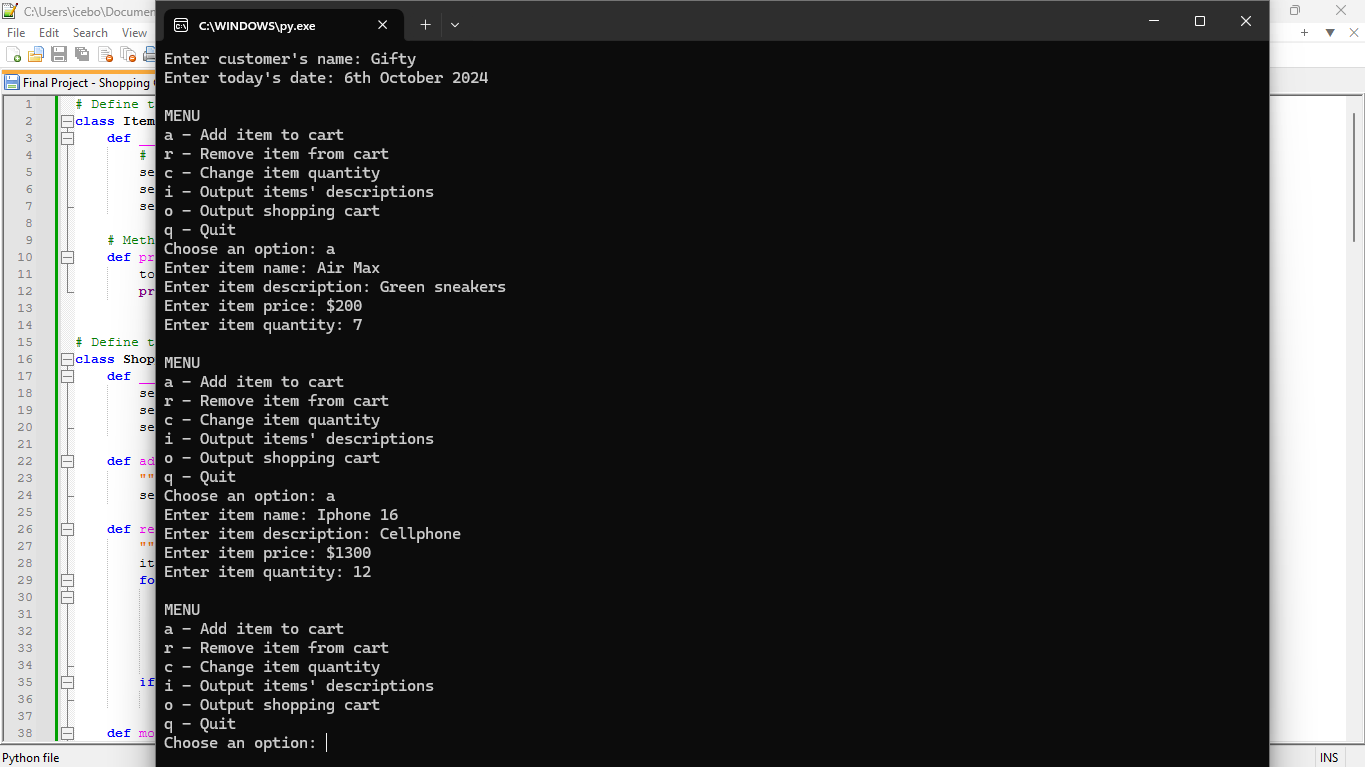
if \_\_name\_\_ == "\_\_main\_\_":

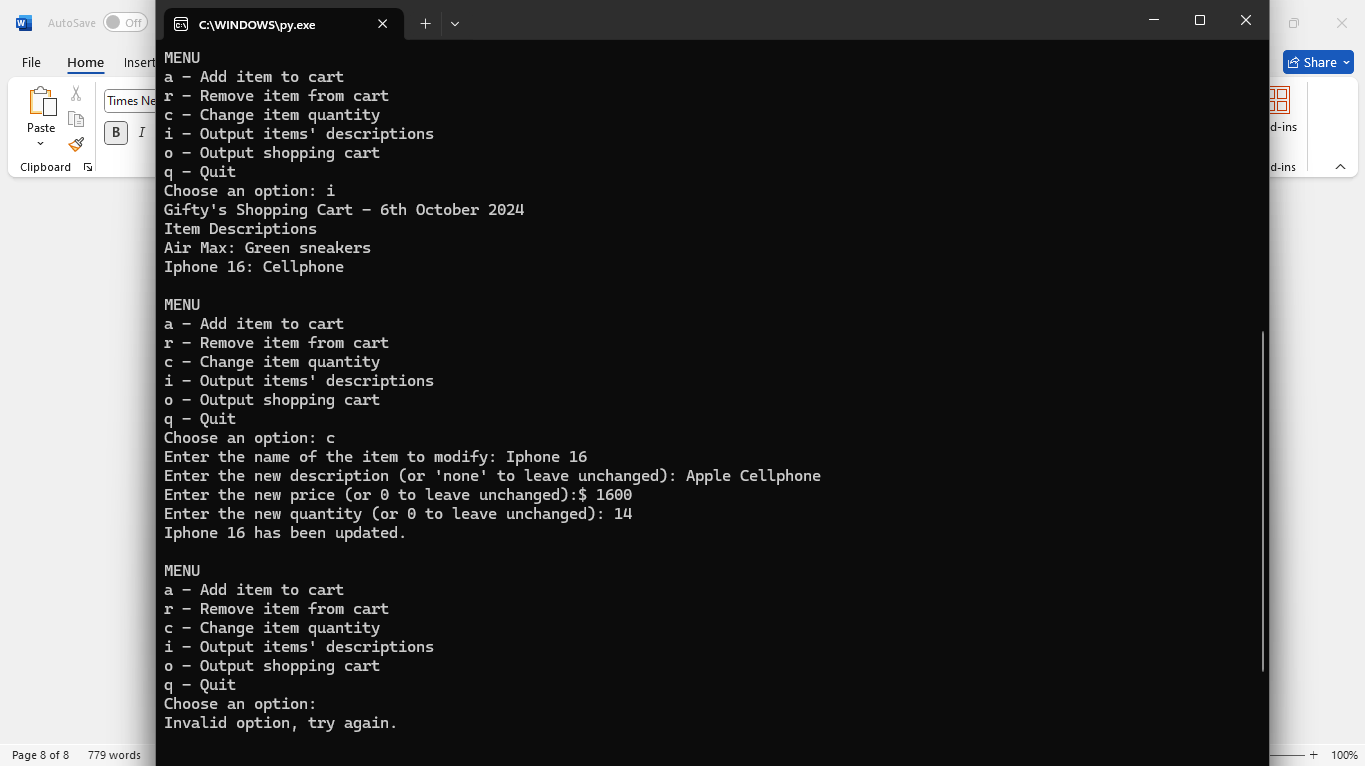
main()

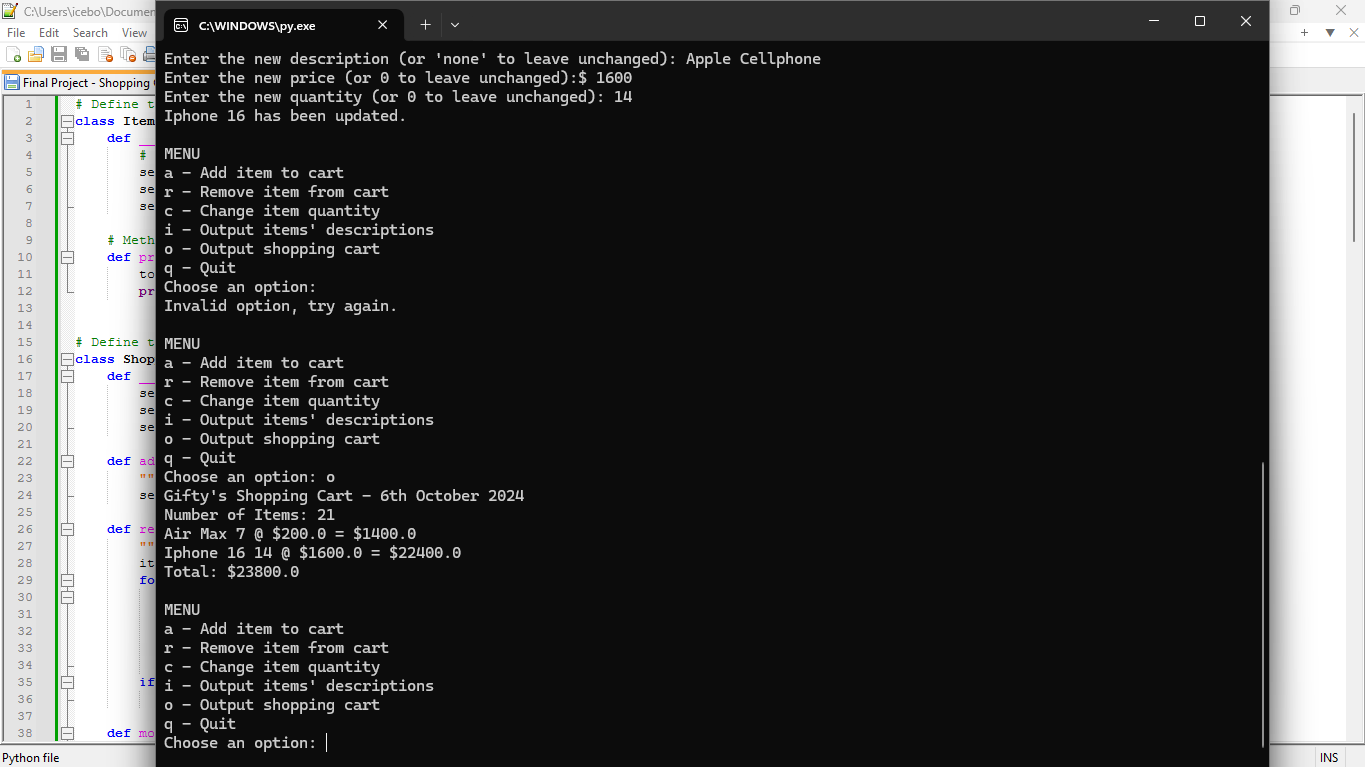
**Screenshots of application running the program:**

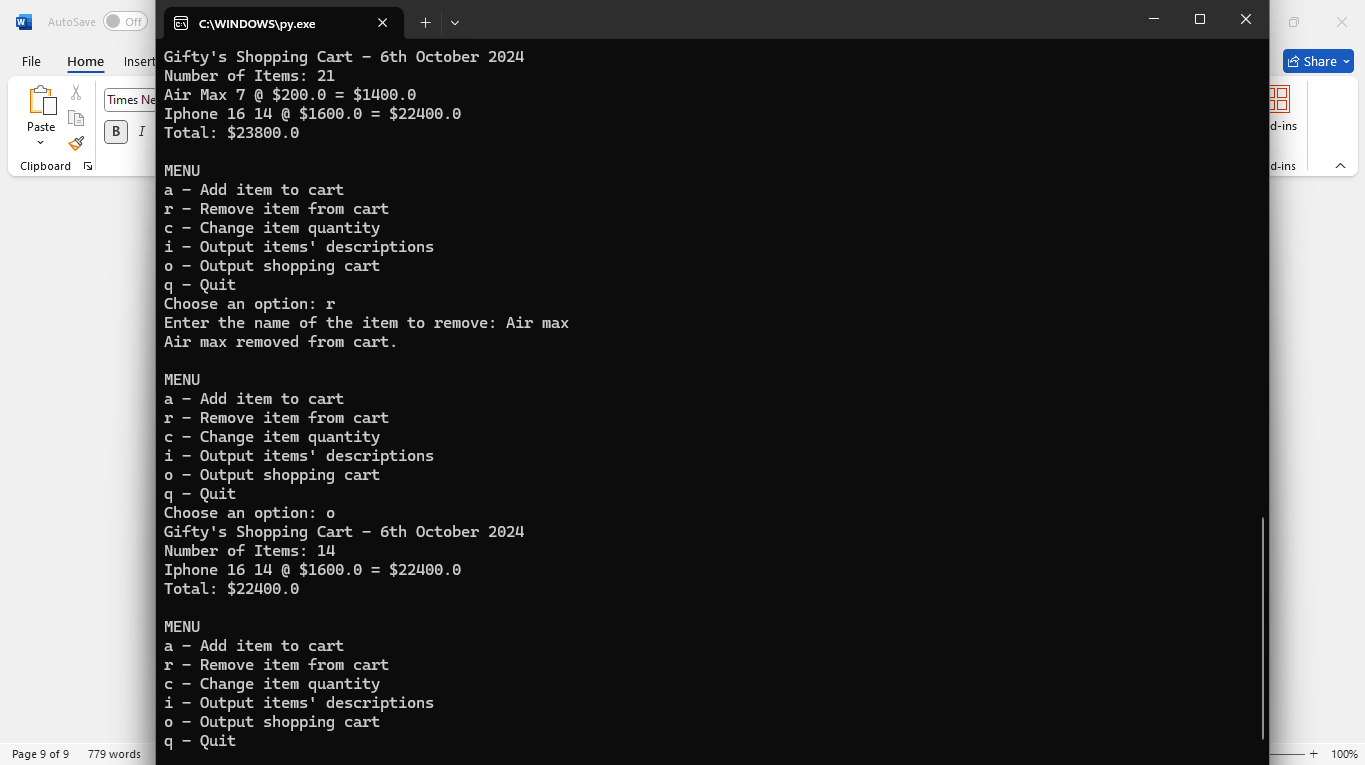
****

****

****

****

****

****

**GIT Repository link:**

<https://github.com/giftataylor/Week-8-final-project/upload>