Adam Tunchay Terminal Application

CODER ACADEMY WEB DEVELOPMENT ACCELERATED 2022

STUDENT - **13537**

Import and Variable Declaration

```python
# Import
import modules

# Food Menu Dictionary
menu = {
    'Chips': 4.50,
    'Nuggets' : 7.00,
    'Jalapeno Poppers' : 8.00,
    'Chicken Burger' : 10.50,
    'Cheeseburger' : 8,
    'Double Cheeseburger' : 11.50,
    'Coke' : 2.00,
    'Sprite' : 2.00,
    'Fanta' : 2.00,
    'Water' : 1.50,
}
```

main.py

```python
# Import
from datetime import datetime, date, timedelta
import main
from prettytable import PrettyTable
import clearing

#Variable Declarations:
order_items = []
total_price = 0
delivery = False
```

modules.py

Main Program Flow

```
if __name__ == '__main__':
    modules.splash()
    modules.clearing.clear()
    print('Hello and welcome to Mcfoo\'s Ordering System')
    print('How can we help you today?\n')
    modules.main_menu()
    print("Thank you for choosing McFoo for your calorie fix today!")
```

Input Validation Function

```
# Input Validation

def get_input(prompt):
    """_This function validates that users have input a valid integer and displays an error message if input is incorrect_

    Args:
        prompt (_str_): _Allows the function to be reused with different prompts throughout application_

    Returns:
        _int_: _returns the users input as a valid integer_
    """
    while True:
        i = (input(prompt))
        if i.isdigit():
            return int(i)
        else:
            print('Invalid input, please enter a menu item number. (Eg - \'1\')')
```

Splash Screen

```python
# Splash Screen for Application
def splash():
    """
    This function prints out the application splash screen art when application is launched
    """
    print('''
  __  __       ______
 |  \/  |     |  ____|
 | \  / |  ___ | |__  ___    ___
 | |\/| | / __||  __|/ _ \  / _ \
 | |  | || (__ | |  | (_) || (_) |
 |_|__|_| \___||_|   \___/  \___/_                _
 |  ____|              | |  / __ \              | |           (_)
 | |__  ___    ___    __| | | |  | | _ __   __| | ___  _ __  _  _ __    __ _
 |  __|/ _ \  / _ \  / _` | | |  | || '__|/ _` | / _ \| '__|| || '_ \  / _` |
 | |  | (_) || (_) || (_| | | |__| || |  | (_| ||  __/| |   | || | | || (_| |
 |_|   \___/  \___/  \__,_|  \____/ |_|   \__,_| \___||_|   |_||_| |_| \__, |
    /\                    | |(_)            | |  (_)                    __/ |
   /  \     _ __   _ __   | | _   ___  __ _ | |_  _   ___   _ __       |___/
  / /\ \   | '_ \ | '_ \  | || | / __|/ _` || __|| | / _ \ | '_ \
 / ____ \  | |_) || |_) | | || || (__| (_| || |_ | || (_) || | | |
/_/    \_\ | .__/ | .__/  |_||_| \___|\__,_| \__||_| \___/ |_| |_|
           | |    | |
           |_|    |_|
''')
    print('Created by Adam Tunchay for Coder Academy Assignment 3 - 2022\n\n')
    print(input('Press Enter to continue'))
```

```
 __  __       ______
|  \/  |     |  ____|
| \  / |  ___ | |__  ___    ___
| |\/| | / __||  __|/ _ \  / _ \
| |  | || (__ | |  | (_) || (_) |
|_|__|_| \___||_|   \___/  \___/_                _
|  ____|              | |  / __ \              | |           (_)
| |__  ___    ___    __| | | |  | | _ __   __| | ___  _ __  _  _ __    __ _
|  __|/ _ \  / _ \  / _` | | |  | || '__|/ _` | / _ \| '__|| || '_ \  / _` |
| |  | (_) || (_) || (_| | | |__| || |  | (_| ||  __/| |   | || | | || (_| |
|_|   \___/  \___/  \__,_|  \____/ |_|   \__,_| \___||_|   |_||_| |_| \__, |
   /\                    | |(_)            | |  (_)                    __/ |
  /  \     _ __   _ __   | | _   ___  __ _ | |_  _   ___   _ __       |___/
 / /\ \   | '_ \ | '_ \  | || | / __|/ _` || __|| | / _ \ | '_ \
/ ____ \  | |_) || |_) | | || || (__| (_| || |_ | || (_) || | | |
/_/    \_\ | .__/ | .__/  |_||_| \___|\__,_| \__||_| \___/ |_| |_|
           | |    | |
           |_|    |_|

Created by Adam Tunchay for Coder Academy Assignment 3 - 2022


Press Enter to continue
```

Main Menu

```python
# Main Menu Function

def menu_display():
    """_Prints out strings that form the main menu_
    """
    print('\n[1] View Food Menu')
    print('[2] Place an Order')
    print('[3] View Current Order')
    print('[4] Finalize Order')
    print('[5] Clear Order')
    print('[0] Quit Application')

def main_menu():
    """_Main menu and flow of application_
    """
    menu_display()
    option = get_input('Enter your selection: ')
    while option != 0:
        if option == 1:
            print_menu()
        elif option == 2:
            new_order()
        elif option == 3:
            view_order()
        elif option == 4:
            pickup_delivery()
            print_receipt()
            file_receipt()
            ready_time()
        elif option == 5:
            clear_order()
        elif option == 0:
            break
        else:
            print('Invalid input, please enter a menu item number. (Eg - \'1\')')
        menu_display()
        option = get_input('Enter your selection: ')
```

```
Hello and welcome to Mcfoo's Ordering System
How can we help you today?


[1] View Food Menu
[2] Place an Order
[3] View Current Order
[4] Finalize Order
[5] Clear Order
[0] Quit Application
Enter your selection:
```

Printing food menu (with index)

```python
# Print food menu items

def print_menu():
    """_Prints out the food menu to the terminal_
    """
    clearing.clear()
    print('Today\'s menu: ')
    i = 1
    for food, price in main.menu.items():
        print (f'{i}.', food, '- $',price)
        i += 1
```

```
Today's menu:
1. Chips - $ 4.5
2. Nuggets - $ 7.0
3. Jalapeno Poppers - $ 8.0
4. Chicken Burger - $ 10.5
5. Cheeseburger - $ 8
6. Double Cheeseburger - $ 11.5
7. Coke - $ 2.0
8. Sprite - $ 2.0
9. Fanta - $ 2.0
10. Water - $ 1.5

[1] View Food Menu
[2] Place an Order
[3] View Current Order
[4] Finalize Order
[5] Clear Order
[0] Quit Application
Enter your selection:
```

Order function

```
Enter your selection: 2

Please enter the menu item number of the food you wish to add to your order:
Enter your selection: 6

Double Cheeseburger added to order.


Would you like to:
[1] Add more items to order
[2] Display current order
[3] Display Food Menu
[4] Back to Main Menu
```

```python
def new_order():
    """_Function to add food menu items to order_items variable_
    _updates total_price when items are added_

    Returns:
        _float_: _total_price_
    """
    print('\nPlease enter the menu item number of the food you wish to add to your order: ')
    global order_items, total_price
    while True:
        order_req = get_input('Enter your selection: ') - 1
        if order_req < len(main.menu):
            for index, (key, value) in enumerate(main.menu.items()):
                if index == order_req:
                    sub_total = value
                    total_price += sub_total
                    order_items.append([key, sub_total])
                    print(f'\n{key} added to order.\n')
                    while True:
                        x = input('\nWould you like to:\n[1] Add more items to order \n[2] Display current order \n[3] Display Food Menu \n[4] Back to Main Menu ')
                        if x == '1':
                            print('What would you like to order next? ')
                            break
                        elif x == '2':
                            view_order()
                            continue
                        elif x == '3':
                            print_menu()
                            continue
                        elif x == '4':
                            return total_price
                        else:
                            print('Invalid input, please enter a menu item number. (Eg - \'1\')')
        else:
            print('Entry not on Menu, please select a valid menu item number. ')
```

View order function

```python
# Display Current Order
def view_order():
    """_Used to display what items have been added to order during use of application_
    """
    if len(order_items) == 0:
        print('\nYou have no items in your order.\n')
    else:
        print('\nYou have ordered:\n')
        for food, price in order_items:
            print(food + ' $' + str(price))
        print(f'Your current total is ${total_price}\n')

# Clear current order
```

```
Enter your selection: 3

You have ordered:

Double Cheeseburger $11.5
Your current total is $11.5


[1] View Food Menu
[2] Place an Order
[3] View Current Order
[4] Finalize Order
[5] Clear Order
[0] Quit Application
Enter your selection:
```

```
Enter your selection: 3

You have no items in your order.
```

Pickup/Delivery Function

```python
# Pickup or delivery function

def pickup_delivery():
    """_User inputs whether their order is pickup or delivery_
    _returns True boolean value to delivery for use in other functions._
    _updates total price with delivery added if applicable_
    """
    global total_price, delivery
    x = get_input('Please select [1] for Delivery or [2] for Pickup. ')
    while True:
        if x == 1:
            total_price += 7.50
            print('\n$7.50 delivery fee has been added to your order.')
            print(f'Your total including delivery is ${total_price}')
            delivery = True
            break
        elif x ==2:
            break
        else:
            print('Invalid input, please enter [1] for Delivery or [2] for Pickup.')
```

```
Please select [1] for Delivery or [2] for Pickup.
```

Clear Order Function

```python
# Clear current order

def clear_order():
    """_Used to clear current order_
    """
    clearing.clear()
    global order_items, total_price, delivery
    order_items = []
    total_price = 0
    delivery = False
    print('Order cleared\n')
```

```
Order cleared


[1] View Food Menu
[2] Place an Order
[3] View Current Order
[4] Finalize Order
[5] Clear Order
[0] Quit Application
Enter your selection:
```

Print Receipt Function

```
# Produce a receipt and output receipt to a text file

def print_receipt():
    """_function to print receipt to terminal_
    _different receipt is printed depending if delivery was selected_
    """
    global table
    table = PrettyTable(['Item', 'Price'])
    clearing.clear()
    for food,price in order_items:
        table.add_row([food,'$' + str(price)])
    table.add_row(['-'* 8,'-' * 8])
    if delivery is True:
        table.add_row(['DELIVERY', '$7.50'])
        table.add_row(['-'* 8,'-' * 8])
        table.add_row(['TOTAL', '$' + str(total_price) ])

    else:
        table.add_row(['TOTAL', '$' + str(total_price)])
    print(table)
    return
```

```
+--------------+----------+
|     Item     |  Price   |
+--------------+----------+
|    Chips     |   $4.5   |
| Cheeseburger |    $8    |
|   --------   | -------- |
|   DELIVERY   |  $7.50   |
|   --------   | -------- |
|    TOTAL     |  $20.0   |
+--------------+----------+
```

Receipt to File Function

```python
# Save Receipt to file

def file_receipt():
    """_function to save the receipt to a text file_
    """
    print('Would you like to store your receipt in a file? \n[1] Yes [2] No ')
    x = get_input('')
    if x == 1:
        with open('receipt.txt', 'w') as f:
            f.write('McFoo Receipt\n')
            f.write('Order Placed: '+ datetime.now().strftime('%B %d, %Y %H:%M:%S') + '\n')
            f.write(str(table))
            f.write('\nThank you for your order!')
            f.write('\nMcFoo Restaurants Australia')
            print('\nYour receipt has been saved as receipt.txt')
```

receipt.txt

```
McFoo Receipt
Order Placed: September 25, 2022 21:30:46
+----------+----------+
|   Item   |  Price   |
+----------+----------+
| -------- | -------- |
| DELIVERY |  $7.50   |
| -------- | -------- |
|  TOTAL   |   $7.5   |
+----------+----------+
Thank you for your order!
McFoo Restaurants Australia
```

Ready Time Function

```
# Using datetime, give an estimate of when the order will be ready or delivered

def ready_time():
    """_used to print pickup or delivery time to the terminal_
    """
    now = datetime.now()
    pickup_time = now + timedelta(minutes=20)
    delivery_time = now + timedelta(minutes=40)
    print('Order placed at ', now.strftime("%H:%M, %d %B %Y"))
    if delivery == True:
        print('Your order will be delivered at approximately', delivery_time.strftime("%H:%M, %d %B %Y\n"))
    else:
        print('Your order will be ready for pickup at', pickup_time.strftime("%H:%M, %d %B %Y\n"))
```

```
Your receipt has been saved as receipt.txt
Order placed at  22:57, 25 September 2022
Your order will be delivered at approximately 23:37, 25 September 2022
```

Build Challenges

- Time! Whilst I feel like I somewhat got there in the end, it was a very stressful week with countless hours poured into the project. There are features that I would have loved to implement but just did not have the time to complete within such a short time span. I got the application running to a level I was happy with in the end. But my documentation and slide deck suffered due to running out of time.
- Picking a project at appropriate level. I originally pitched a Tic Tac Toe idea, which I felt that I would not have been able to get working within the short time span. Unfortunately, I spent a few days trying to work it out, which then hindered the production of my end application.
- OOP fail – I was eager to use OOP to create my menu and functions, but I spent too much time researching without producing results and ended up ditching the idea and using a different data structure in the end. This also led to a waste of hours that I could've put into the project.

Ethical Issues

- No real ethical issues with this project.

Future Implementations

- I wanted to implement the option for the user to enter dietary requirements into the application and be presented with menu items that are appropriate for them.
- Customer information – User can enter name, address etc. to be stored for delivery.

Favourite Parts

- I had a lot of trouble getting started with this project. It was very daunting starting with a blank document and I really didn't know where to begin. Once I broke it down and began coding, it was really rewarding seeing the application come together. When working through content in class and challenges online, it's somewhat hard to picture how the code is used in a real application. This was my first time creating an app and I really enjoyed the final product!
- The feeling you get after researching for hours to get one little line of code to work properly. You are so frustrated when it is broken for hours, but the feeling you get once it finally executes without errors is a feeling that I want more of, more often!