Carvana Vehicle Purchasing Program

PYTHON PROJECT (CHAPTERS 1–8)

Program Overview

```
68  def main():
69      print("==================================")
70      print("       WELCOME TO CARVANA         ")
71      print("==================================\n")
```

- Carvana-style car purchasing system
- Built using Python
- Fully interactive
- Uses external text files
- Ends with a purchase receipt or restart

```
299     # Run program
300     main()
```

FAQ Feature

- User can type FAQ at any time
- Displays help information
- Makes the program user-friendly
- Works at every input prompt

```python
def show_faq():
    # Opens and displays the FAQ file
    faq_file = open('faq.txt', 'r')
    faq_contents = faq_file.read()
    faq_file.close()
    print("\n" + faq_contents + "\n")
```

```python
def get_input_with_faq(prompt):
    # Asks for input and checks for FAQ manually
    user_input = input(prompt)

    while equals_faq(user_input):
        show_faq()
        user_input = input(prompt)

    return user_input
```

Input Validation

- Ensures the user enters valid input
- Prevents program crashes
- Checks numeric values
- Confirms selected car exists

```python
def is_number(value):
    # Validates that the input is numeric (no .isdigit() allowed)
    try:
        int(value)
        return True
    except:
        return False

```

```python

    while not is_number(user_input):
        print("Please enter a number.\n")
        user_input = get_input_with_faq('Which number car would you like to view? ')

    choice = int(user_input)

```

Viewing the Catalog

- Reads data from catalog.txt
- Displays a numbered list of cars
- Easy-to-read format
- User selects a car by number

```
83
84          # Read catalog
85          catalog_file = open('catalog.txt', 'r')
86          catalog_lines = catalog_file.readlines()
87          catalog_file.close()
88
89          # Print catalog with numbers
90          number = 1
91          for line in catalog_lines:
92              clean = line.strip()
93              if clean != "":
94                  print(str(number) + ". " + clean)
95                  number += 1
96
```

```
100         # Ask which car
101         user_input = get_input_with_faq('Which number car would you like to view? ')
102
103         while not is_number(user_input):
104             print("Please enter a number.\n")
105             user_input = get_input_with_faq('Which number car would you like to view? ')
106
107         choice = int(user_input)
108
```

Vehicle Specifications

Reads data from specs.txt

Displays full vehicle details

Linked to catalog selection

User chooses whether to continue

```
108
109         # Read specs
110         specs_file = open('specs.txt', 'r')
111         specs_lines = specs_file.readlines()
112         specs_file.close()
113
114         # Validate
115         if choice < 1 or choice > len(specs_lines):
116             print("\nThat number is not in the catalog. Please restart the program.\n")
117             return
118
119         chosen_specs = specs_lines[choice - 1].strip()
120
121         print("\nHere are the details for that car:\n")
122         print(chosen_specs)
123
```

Purchasing Process

- Trade-in estimate
- Income & down payment check
- Shipping options
- Multiple payment methods

```
147         import random
148         trade_value = random.randint(2000, 15000)
149
150         print("\nCarvana's trade-in estimate for your:")
151         print(year, make, model)
152         print("Trade-in Offer: $", trade_value, "\n", sep='')
153
```

Price & Calculations

- Extracts vehicle price from specs
  - Calculates sales tax
  - Adds shipping if selected
- Calculates totals and payments

```
182
183         # PRICE EXTRACTION — no methods allowed
184         price = extract_price(chosen_specs)
185
```

```
199
200         tax_rate = 0.08875
201         tax_amount = int(price * tax_rate)
202
```

Purchase Summary

Selected vehicle

Base price

Tax & shipping

Payment method

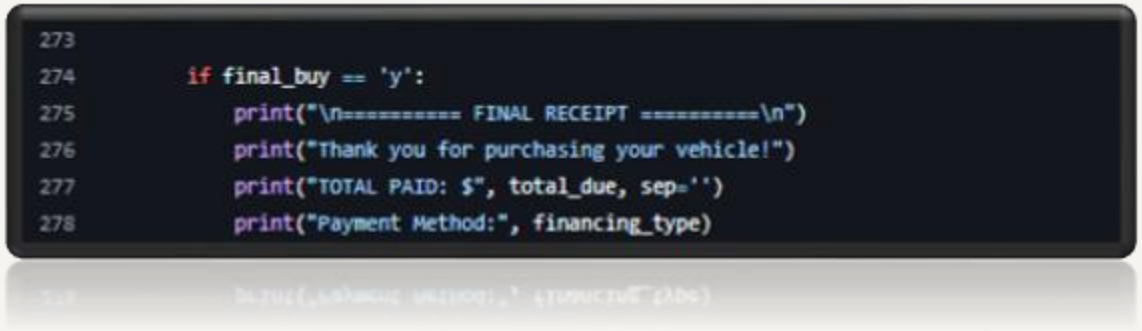
Monthly payment (if financing)

Total amount due

```
# PURCHASE SUMMARY
print("\n--- PURCHASE SUMMARY ---\n")

print("Car Selected: " + chosen_specs)
print("-------------------------------------")
print("Base Price: $", price, sep='')
print("NY Sales Tax (8.875%): $", tax_amount, sep='')
print("Trade-in Value: -$", trade_value, sep='')
print("Shipping Cost: $", shipping_cost, sep='')
print("-------------------------------------")
print("Payment Type:", financing_type)

if financing_type != "Cash Purchase":
    print("Estimated Interest Rate:", interest_rate * 100, "%")
    print("Loan Term:", finance_months, "months")
    print("Estimated Monthly Payment: $", monthly_payment, sep='')

print("-------------------------------------")
print("TOTAL DUE: $", total_due, sep='')
print("=====================================\n")
```

Final Receipt

- User confirms purchase
- Displays final receipt
- Shows pickup or delivery info
- Option to return to catalog



```
273
274         if final_buy == 'y':
275             print("\n========== FINAL RECEIPT ==========\n")
276             print("Thank you for purchasing your vehicle!")
277             print("TOTAL PAID: $", total_due, sep='')
278             print("Payment Method:", financing_type)
```

Python Concepts Used

- File input
- Functions
- Loops
- Conditional statements
- Input validation
- String processing
- Basic math calculations

Conclusion

- Complete car-buying simulation
    - Beginner-friendly design
    - Organized and easy to follow
- Demonstrates core Python skills