CS212 Python Final Computer Project

Assignment Overview:

This assignment focuses on the design, implementation and testing of a Python program to solve the problem described below. **DUE: Completed program no later than 5:30 PM on Monday, December 3, 2018**.

Assignment Specifications

1. The program will compute and display information for a company which rents vehicles to its customers. For a specified customer, the program will compute and display the amount of money charged for that customer's vehicle rental.

2. The program will repeatedly prompt the user to enter the following four items for a given customer (in the specified order). It will halt when the user enters "Q" (or "q") instead of a classification code:

- The customer's classification code (a character)
- The number of days the vehicle was rented (an integer)
- The vehicle's odometer reading at the start of the rental period (an integer)
- The vehicle's odometer reading at the end of the rental period (an integer)

3. Create a user-defined function in your program to compute the amount of money that the customer will be billed, based on the customer's classification code, number of days in the rental period, and number of miles driven.

The program will recognize both upper case and lower case letters for the following classification codes:

- Code 'B' (budget) base charge: $40.00 for each day mileage charge: $0.25 for each mile driven
- Code 'D' (daily) base charge: $60.00 for each day mileage charge: no charge if the average number of miles driven per day is 100 miles or less; otherwise, $0.25 for each mile driven above the 100 mile per day limit.
- Code 'W' (weekly) base charge: $190.00 for each week (or fraction of a week) mileage charge: no charge if the average number of miles driven per week is 900 miles or less; $100.00 per week if the average number of miles driven per week exceeds 900 miles but does not exceed 1500 miles; otherwise, $200.00 per week plus $0.25 for each mile driven above the 1500 mile per week limit.

The amount billed to the customer is the sum of the base charge and the mileage charge.

4. The program will compute the number of miles driven by the customer during the rental period. The odometer readings are taken from an odometer which has six digits and records tenths of a mile.

5. For each customer, the program will display a summary with the following information:

- The customer's classification code
- The number of days the vehicle was rented
- The vehicle's odometer reading at the start of the rental period
- The vehicle's odometer reading at the end of the rental period
- The number of miles driven during the rental period (will be rounded to one fractional digit).
- The amount of money billed to the customer for the rental period (will be displayed with a dollar sign and will be rounded to two fractional digits for example, $125.99 or $43.87).

6. The program will detect, report and recover from invalid classification codes. When an invalid classification code is detected, the program will display an error message. It will display the summary for that customer, with the amount billed to the customer set to zero.

7. The program will assume that all other user inputs are valid and correct. That is, the program will not check the number of days or odometer readings for validity.

Assignment Notes

1. As stated above, the odometer's dial has six digits and records tenths of a mile. For example, if the beginning reading was 100003 and the ending reading was 100135, then the customer drove 13.2 miles during the rental period.

2. Since the odometer's dial only has six digits, the reading at the end of the rental period may be less than the reading at the beginning of the billing period. For example, if the beginning reading was 999997 and the ending reading was 000005, then the customer drove 0.8 miles during the rental period.

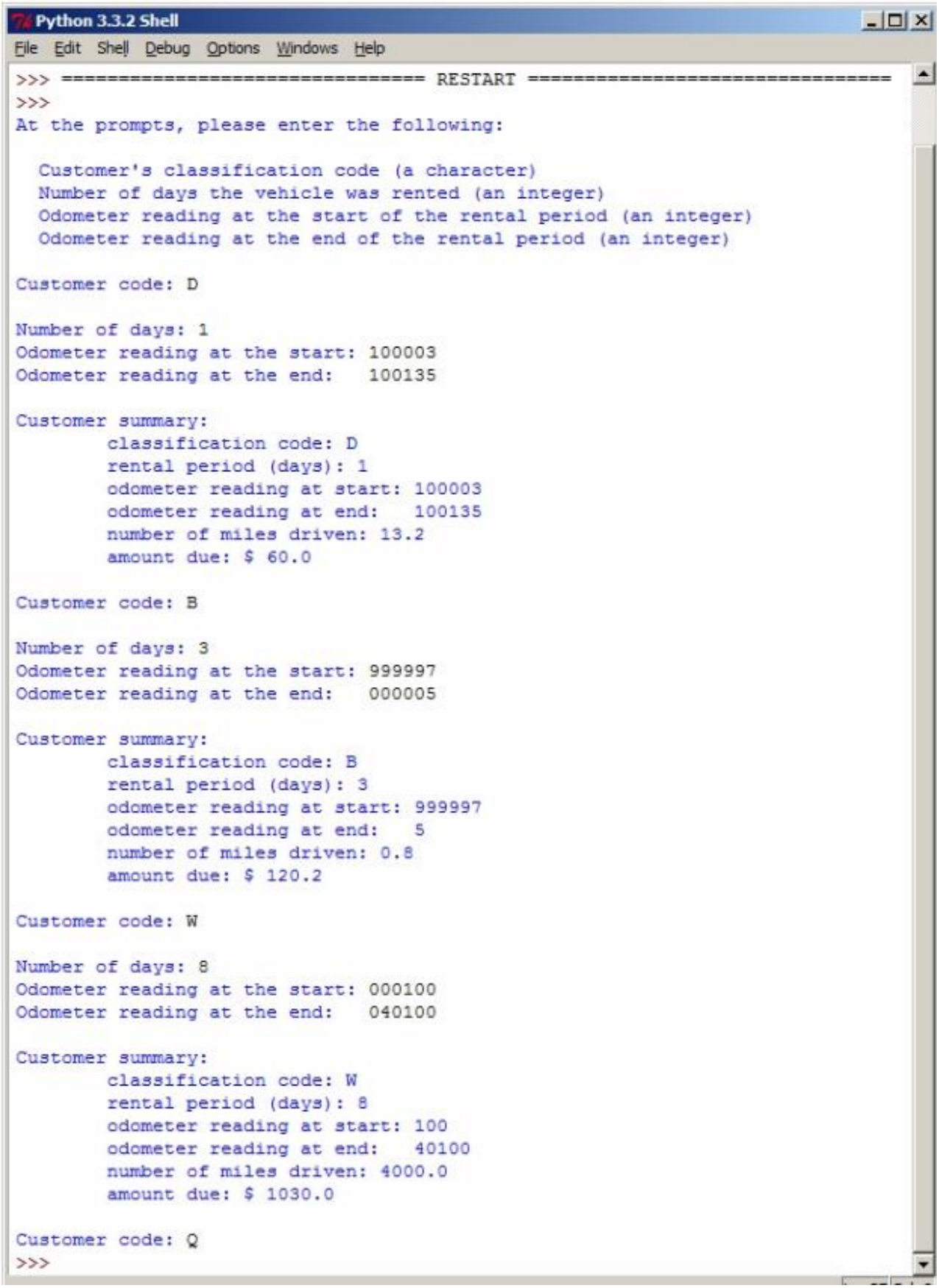
3. Be sure to prompt the user for the four inputs in the correct order. And, your program cannot prompt the user for any supplementary inputs.

4. Note that your program should gracefully handle invalid classification codes, but is not expected to handle other user-supplied inputs which are invalid.

5. Use the sample data in the graphic below to test your program.

Suggested Procedure

1. Break the problem down into smaller steps and start with a simple version of the program.
2. You may want to solve the problem using pencil and paper first. You cannot write a program until you have figured out how to solve the problem.
3. Cycle through the steps to incrementally develop your program.
4. Edit your program to add new capabilities.
5. Run the program and fix any errors.
6. You must work on your own.


Python 3.3.2 Shell
File Edit Shell Debug Options Windows Help
>>> ========== RESTART ==========
>>>
At the prompts, please enter the following:
Customer's classification code (a character)
Number of days the vehicle was rented (an integer)
Odometer reading at the start of the rental period (an integer)
Odometer reading at the end of the rental period (an integer)
Customer code: D
Number of days: 1
Odometer reading at the start: 100003
Odometer reading at the end: 100135
Customer summary:
classification code: D
rental period (days): 1
odometer reading at start: 100003
odometer reading at end: 100135
number of miles driven: 13.2
amount due: $ 60.0
Customer code: B
Number of days: 3
Odometer reading at the start: 999997
Odometer reading at the end: 000005
Customer summary:
classification code: B
rental period (days): 3
odometer reading at start: 999997
odometer reading at end: 5
number of miles driven: 0.8
amount due: $ 120.2
Customer code: W
Number of days: 8
Odometer reading at the start: 000100
Odometer reading at the end: 040100
Customer summary:
classification code: W
rental period (days): 8
odometer reading at start: 100
odometer reading at end: 40100
number of miles driven: 4000.0
amount due: $ 1030.0
Customer code: Q
>>>