CSE 231 **Fall 2020**

Project #2

This assignment focuses on the design, implementation and testing of a Python program which uses control structures to solve the problem described below.
It is worth 20 points (2% of course grade) and must be completed no later than **11:59 PM on Monday, September 21**.

Assignment Overview
(learning objectives)
This assignment will give you more experience on the use of:

- integers (int)
- conditionals
- iteration
- input/output

Assignment Specifications

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.

1. The program should start by asking the user if he wants to continue. The answer is `'Y'` for yes or `'N'` for no. You can check the value with Boolean expressions such as

```
answer == 'Y'
```

or

```
answer != 'Y'.
```

The basic structure of the main loop is

```
Prompt to see if the user wants to continue
While the value is 'Y':
    All your other code goes here
    Prompt to see if the user wants to continue
```

2. It will then continue to prompt the user to enter the following four items for a given customer (in the specified order):

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

It will then process that customer information and display the results. At the end, the program should ask the user if he wants to process another request and it will keep asking until the user enter 0.

3. The program will 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 only recognize both upper case letters for the 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. (Note: if you rent for 8 days that is more than one week so the base rate is for 2 weeks – hint: use `math.ceil()`. Also, you are to take the total miles and subtract `1500*weeks` to get the miles you will be charging for.)

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:

a. The customer's classification code
b. The number of days the vehicle was rented
c. The vehicle's odometer reading at the start of the rental period
d. The vehicle's odometer reading at the end of the rental period
e. The number of miles driven during the rental period
f. The amount of money billed to the customer for the rental period

All output will be appropriately labeled and formatted. The number of miles driven will be rounded to one fractional digit. The amount of money billed will be displayed with a dollar sign and will be rounded to two fractional digits (for example, $125.99 or $43.87). Note that we do not have the ability yet to fine tune the output so if cents end in zero that final zero will not be displayed—that is fine for this project. We provide a file `strings.txt` with the strings we used to make it easier for you to match our output.

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 and re-prompt until a correct classification code is entered.

Hint: use a `while` loop that will loop until a correct classification code is entered. Here is *pseudo code* of that loop:

```
Prompt for a value
While the value is not correct:
    Print an error message
    Prompt for a value
```

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. However, as noted below you may encounter an ending odometer reading that is less than the beginning reading and you need to handle that correctly.

Assignment Deliverable

The deliverable for this assignment is the following file:

`proj02.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via the **Mimir** before the project deadline.

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 is fixed with only 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. You need to handle that arithmetic correctly.

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

4. The structure of the main loop is as follows (in pseudo-code):

```
Prompt for an input   # for example, answer = input(…
while Boolean:   # Boolean expression uses input such as “answer”
    # all your code goes here
    Prompt for an input  (same statement used above)
```

5. It isn’t necessary for the user to type leading zeroes when entering odometer readings. However, it doesn’t hurt anything, either.

6. Coding Standards 1 – 6 will be enforced.

7. You are not allowed to use functions or advanced data structures such as lists, dictionaries, etc. That is, things we haven’t covered yet.

8. *Beware* of this common programming error: you may be thinking of a Boolean expression this way:

```
x == ‘B’ or ‘D’ or ‘W’
```

However, that evaluates to `True` for any value of `x` because any **non-empty** string evaluates to `True` so that Boolean expression is equivalent to

```
x = ‘B’ or True or True
```

whose value is `True` for any value of `x`.
The logically correct way to write that expression is

```
x == ‘B’ or x == ‘D’ or x == ‘W’
```

If you want the negation of that, you can do the following:

```
not(x == ‘B’ or x == ‘D’ or x == ‘W’)
```

Note that this next expression does **not** have the same value!

```
x != ‘B’ or x != ‘D’ or x != ‘W’)
```

Understanding why not is a useful exercise. (For the curious lookup DeMorgan’s Law.)

Suggested Procedure

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step is best done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python statements, you must work on your own.

- Write a simple version of the program. Run the program and track down any errors.

- Use the **Mimir** system to turn in the first version of your program.

- Cycle through the steps to incrementally develop your program:
  - Edit your program to add new capabilities.
  - Run the program and fix any errors.

- Use the **Mimir** system to submit your final version.

- Be sure to log out when you leave the room, if you're working in a public lab.

Test 1

```
Welcome to car rentals.

At the prompts, please enter the following:
     Customer's classification code (a character: BDW)
     Number of days the vehicle was rented (int)
     Odometer reading at the start of the rental period (int)
     Odometer reading at the end of the rental period (int)

Would you like to continue (Y/N)? N
Thank you for your loyalty.
```

Test 2

```
Welcome to car rentals.

At the prompts, please enter the following:
     Customer's classification code (a character: BDW)
     Number of days the vehicle was rented (int)
     Odometer reading at the start of the rental period (int)
     Odometer reading at the end of the rental period (int)

Would you like to continue (Y/N)? Y

Customer code (BDW): 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

Would you like to continue (Y/N)? Y

Customer code (BDW): D

Number of days: 4
Odometer reading at the start: 101010
Odometer reading at the end:   108200

Customer summary:
     classification code: D
     rental period (days): 4
     odometer reading at start: 101010
     odometer reading at end:   108200
     number of miles driven:  719.0
     amount due: $ 319.75

Would you like to continue (Y/N)? Y

Customer code (BDW): D

Number of days: 2
Odometer reading at the start: 002000
Odometer reading at the end:   004000

Customer summary:
     classification code: D
     rental period (days): 2
     odometer reading at start: 2000
     odometer reading at end:   4000
     number of miles driven:  200.0
     amount due: $ 120.0

Would you like to continue (Y/N)? Y

Customer code (BDW): 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

Would you like to continue (Y/N)? N
Thank you for your loyalty.
```

Test 3

```
Welcome to car rentals.

At the prompts, please enter the following:
     Customer's classification code (a character: BDW)
     Number of days the vehicle was rented (int)
     Odometer reading at the start of the rental period (int)
     Odometer reading at the end of the rental period (int)

Would you like to continue (Y/N)? Y

Customer code (BDW): D

Number of days: 3
Odometer reading at the start: 002000
Odometer reading at the end:   002100

Customer summary:
     classification code: D
     rental period (days): 3
     odometer reading at start: 2000
     odometer reading at end:   2100
     number of miles driven:  10.0
     amount due: $ 180.0

Would you like to continue (Y/N)? Y

Customer code (BDW): D

Number of days: 8
Odometer reading at the start: 000200
Odometer reading at the end:   010000

Customer summary:
     classification code: D
     rental period (days): 8
     odometer reading at start: 200
     odometer reading at end:   10000
     number of miles driven:  980.0
     amount due: $ 525.0

Would you like to continue (Y/N)? N
Thank you for your loyalty.
```

Test 4

```
Welcome to car rentals.

At the prompts, please enter the following:
      Customer's classification code (a character: BDW)
      Number of days the vehicle was rented (int)
      Odometer reading at the start of the rental period (int)
      Odometer reading at the end of the rental period (int)

Would you like to continue (Y/N)? Y

Customer code (BDW): 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

Would you like to continue (Y/N)? Y

Customer code (BDW): W

Number of days: 6
Odometer reading at the start: 000100
Odometer reading at the end:   001000

Customer summary:
      classification code: W
      rental period (days): 6
      odometer reading at start: 100
      odometer reading at end:   1000
      number of miles driven:  90.0
      amount due: $ 190.0

Would you like to continue (Y/N)? N
Thank you for your loyalty.
```

Test 5

```
Welcome to car rentals.
```

```
At the prompts, please enter the following:
     Customer's classification code (a character: BDW)
     Number of days the vehicle was rented (int)
     Odometer reading at the start of the rental period (int)
     Odometer reading at the end of the rental period (int)

Would you like to continue (Y/N)? Y

Customer code (BDW): x

     *** Invalid customer code. Try again. ***

Customer code (BDW): y

     *** Invalid customer code. Try again. ***

Customer code (BDW): W

Number of days: 14
Odometer reading at the start: 000100
Odometer reading at the end:   001000

Customer summary:
     classification code: W
     rental period (days): 14
     odometer reading at start: 100
     odometer reading at end:   1000
     number of miles driven:  90.0
     amount due: $ 380.0

Would you like to continue (Y/N)? Y

Customer code (BDW): W

Number of days: 15
Odometer reading at the start: 000000
Odometer reading at the end:   028000

Customer summary:
     classification code: W
     rental period (days): 15
     odometer reading at start: 0
     odometer reading at end:   28000
     number of miles driven:  2800.0
     amount due: $ 870.0

Would you like to continue (Y/N)? Y
```

```
Customer code (BDW): W

Number of days: 10
Odometer reading at the start: 000100
Odometer reading at the end:   090000

Customer summary:
     classification code: W
     rental period (days): 10
     odometer reading at start: 100
     odometer reading at end:   90000
     number of miles driven:  8990.0
     amount due: $ 2277.5

Would you like to continue (Y/N)? N
Thank you for your loyalty.
```

Grading Rubric

```
Computer Project #02                             Scoring Summary
General Requirements:
  ( 3 pts) Coding Standard 1-6
             (descriptive comments, mnemonic identifiers, format,
     etc...)

Implementation:
 (2 pts) Test Case 1:
 (3 pts) Test Case 2:
 (3 pts) Test Case 3:
 (4 pts) Test Case 4:
 (5 pts) Test Case 5:
```