

CECS 174 - Project 2
“Water Conservation - Water Utility Billing”
Due date: 03/14/22

Team: Joshua and Alfredo

Student Name: Joshua Correa
Student ID: 029196984

I certify that this submission is my original work

Joshua Correa

<< Add one for each team member, submit one report as a team >>

Student Name: Alfredo Regla
Student ID: 027962816

I certify that this submission is my original work

Alfredo Regla

Project Report: Programming Project 2 - “Water Conservation - Water Utility Billing”

1. Goal:

Our goal was to create a program that would compute and display the billing information for water utilities. The water utility is categorized into 3 distinct groups: Industrial, Commercial, and Residential. These 3 groups have different ways of charging water utility. It was our job to write a program that would accommodate these differences.

2. Problem Description:

We were required to implement a system that would take the customer code and readings (start and end) from someone and give them the total for their Water Utility Bill using different math operators and calculating their used gallons.

3. Program Description:

a.

The inputs we used for the problem were **customer_code**, **start_reading**, and **end_reading**. The **customer_code** allowed us to find out which district the user was a part of: Residential, Industrial, or Commercial. The **start_reading** and **end_reading** is used to find out the **used_gallons** which can be used to find out the **to_bill**. Once we have the readings we find the **used_gallons** variable. Then

we do the specified calculations (decided by the **customer_code**) to the **used_gallons** which outputs the **to_bill**.

b.

We used multiple test cases that test the program's ability to detect valid inputs and produce correct outputs. Most of the test cases we used were from the project sheet which outlined the problem; however, we needed to come up with a test case for Industrial since none of the test cases provided covered that group. We did:

Enter customer code (R, C, or I):I

Enter beginning reading (between 0 and 999999999):4

Enter end reading (between 0 and 999999999):999999999

Customer code: I

Beginning reading value in gallons and tenths of gallon 0.4

Ending reading value in gallons and tenths of gallon 99999999.9

Gallons of water used: 99999999.5

Amount Billed\$: 24500.00

Another Test Case we tried for Residential is:

Enter customer code (R, C, or I):R

Enter beginning reading (between 0 and 999999999):500

Enter end reading (between 0 and 999999999):400

Customer code: R

Beginning reading value in gallons and tenths of gallon 50.0

Ending reading value in gallons and tenths of gallon 40.0

Gallons of water used: 99999990.0

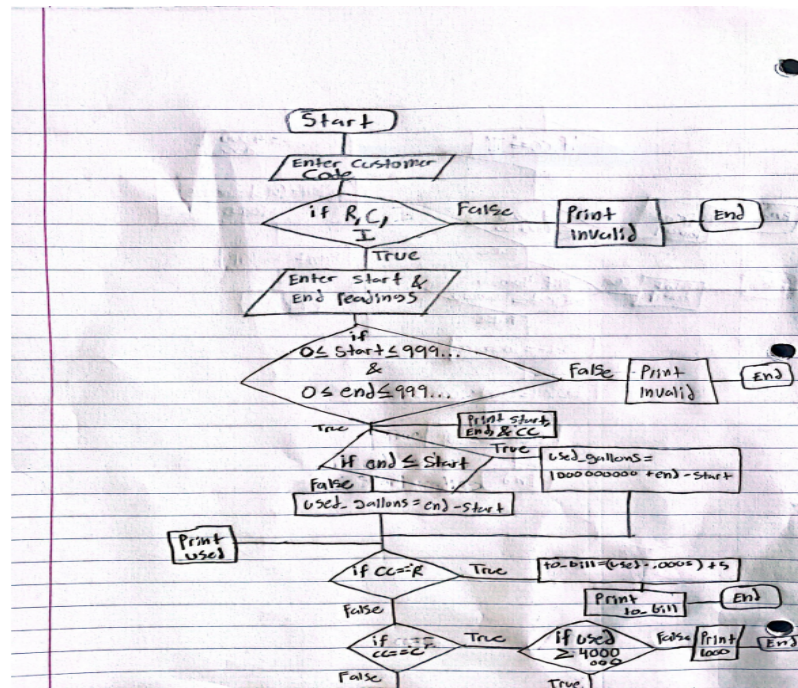
Amount Billed\$: \$50005.00

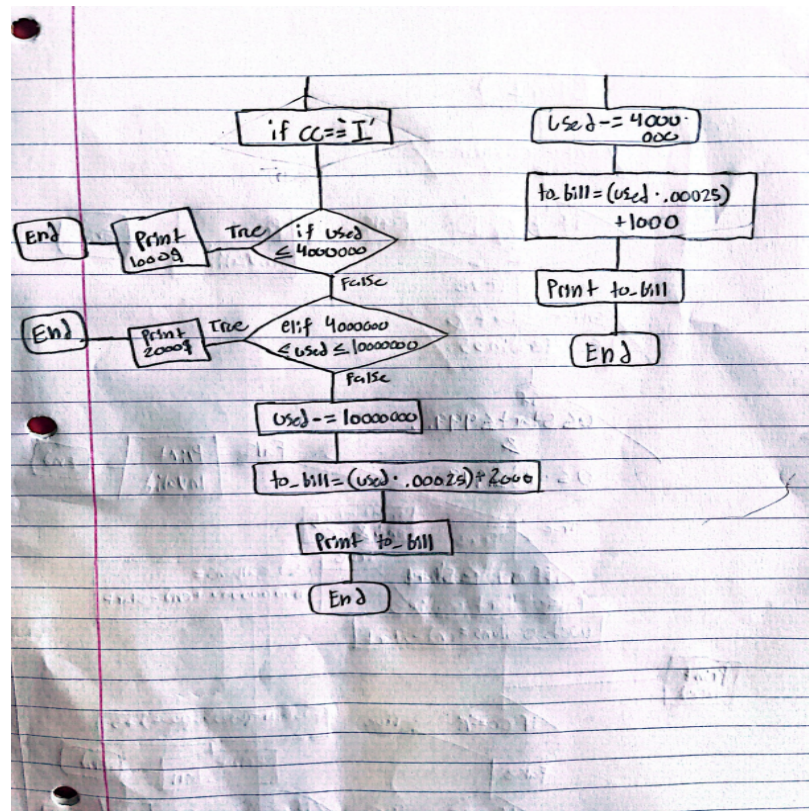
c.

First we ask the user to input their customer code, and then we validate the code. If the code is valid the user continues, if not the code ends. Once the input is validated we ask the user to input their start and end readings. Again if the readings are invalid the code ends, if the readings are valid the code prints out the imputed readings and customer code. The code then compares the readings to see if one is bigger than the other. If the end is bigger than the start we simply subtract the end and start getting used gallons. However if the end is bigger than the start we must add 10,000,000 and then subtract the end and start to get used gallons. Once we have used gallons the code prints it then continues. If the user's customer code was R then we multiply the used gallons by .0005 and then add 5 to get the total bill. if the customer code was C we check if the used gallons is greater than or equal to 4000000. If that statement is false we print 1000 as

the total bill. If the statement is true then we subtract 4000000 from used gallons and then multiply the used gallons by .00025 and add 1000 getting the total bill. If the customer code was I we check if the used gallons is less than or equal to 4000000. If that statement is true we simply print 1000 as the total bill. If that statement is false we check if the used gallons is between 4000000 and 10000000. If that is true we print 2000 as the total bill. If both of those statements were false we subtract 10000000 from the used gallons and then multiply it by .00025 adding 2000 to get the total bill. Once we get the total bill from any of the statements we print it for the user and the code ends.

d.





4. Program Implementation:

- We only used the data type 'int' because all the numbers involved were integers, and the time we used it was when we asked the user for start_reading and end_reading and the rest was calculated with operations. The operations we used consisted of checking if the start_reading values were less than the max and end_reading as well. We also checked if end_reading was less than or equal to start_reading. After all of that was checked we use multiplication to multiply the used_gallons and end_reading by .1. The last one we used is addition to get the amount of gallons used. The only functions used were if, elif, and else statements to comply with everything together. We check if customer code was equal to either R,C,I and would run those if statements and if anything else was run it would go through the else.
- The most challenging part of this program was for sure the beginning, we couldn't figure out how to being the code when the user inputted either R,C, or I. Everything we tried each letter it wouldn't run. We fixed it by using or statements and asking the inputs for the water readings after.

- c. The most fun/straightforward thing to implement was after a letter was inputted you would have to hard code all the mathematics and was also the most fun because it was straightforward and didn't require any critical thinking.
- d. We tested every time of scenario, inputting a wrong letter, or giving wrong values, and tested each letter and random inputs. We did this to make sure no input could break the code and all the outputs looked good.
- e. The program handles bad input very well. Any other input besides R, C, or I goes straight to the else statement which then prints to the user "Invalid input(customer code)".
- f.
 - i. N/A
 - ii. After we finished everything we only had one bug, the bug was that when the user inputted a letter that wasn't either R,C, or I the code would still run and ask for the water gallons, and the problem was that our if statements that ran the code were outside the initial question, so to fix that we combined all the if statements into one for each letter.

5. Conclusion:

- a. What went well is that we collaborated on coming up with ideas to make the code work, We split up the work where someone would do the mathematical part of the code so multiplying and someone else would work on making sure everything ran smooth so the print statements, f strings, and x. This worked well because anytime we had a problem with each other's code we would just work on it together then go back.
- b. We spent a lot of time working on the beginning inputs because when the user inputted a letter it wouldn't run our code, so maybe instead of just trying to figure it out on our own we should have tried to brainstorm together and figure out how we can run it.
- c. The project was pretty straight forward but to make it clearer maybe add background info before we start, like the flowchart was helpful but doing it before maybe can help even more.
- d. The only suggestion I have is that maybe to add context before we do the project, so maybe people won't have a hard time brainstorming while coding

Appendix: ↓

Project 2 Source Code

```
main.py ×
1  customer_code = input('Enter customer code (R, C, or I):')
2
3  if (customer_code == 'R' or customer_code == 'C' or customer_code == 'I'):
4      start_reading = int(
5          input('Enter beginning reading (between 0 and 999999999):'))
6      end_reading = int(input('Enter end reading (between 0 and 999999999):'))
7      if (0 <= start_reading <= 999999999 and 0 <= end_reading <= 999999999):
8          if (end_reading < start_reading):
9              print("Customer code:", customer_code)
10             print(
11                 f'Beginning reading value in gallons and tenths of gallon {start_reading *.1}'
12             )
13             print(
14                 f'Ending reading value in gallons and tenths of gallon {end_reading *.1}'
15             )
16             used_gallons = 1000000000 - start_reading
17             used_gallons *= .1
18             end_reading *= .1
19             used_gallons = end_reading + used_gallons
20             print("Gallons of water used: ", used_gallons)
21     else:
22         start_reading *= .1
23         end_reading *= .1
24         print("Customer code:", customer_code)
25         print(
26             f'Beginning reading value in gallons and tenths of gallon {start_reading}'
27         )
28         print(
29             f'Ending reading value in gallons and tenths of gallon {end_reading}'
30         )
31         used_gallons = end_reading - start_reading
32         print("Gallons of water used:", used_gallons)
33
34     if (customer_code == 'R'):
35         to_bill = (used_gallons * .0005) + 5
36         print(f'Amount Billed$: ${to_bill:.2f}')
37
38     if (customer_code == 'C'):
39         if (used_gallons >= 4000000):
40             used_gallons -= 4000000
41             to_bill = (used_gallons * .00025) + 1000
42             print(f'Amount Billed$: ${to_bill:.2f}')
43         else:
44             print('Amount Billed$: $1000')
45
46     if (customer_code == 'I'):
47         if (used_gallons <= 4000000):
48             print('Amount Billed$: $1000')
49         elif (4000000 < used_gallons <= 10000000):
50             print('Amount Billed$: $2000')
51         else:
52             used_gallons -= 10000000
53             to_bill = (used_gallons * .00025) + 2000
54             print(f'Amount Billed$: ${to_bill:.2f}')
55     else:
56         print('Invalid input(beginning or ending reading value is out of the range)')
57
58
59 else:
60     print('Invalid input(customer code)')
61
```

Project 2 Output for Customer Code (I)

Console Shell

```
Enter customer code (R, C, or I):I
Enter beginning reading (between 0 and 999999999):4
Enter end reading (between 0 and 999999999):999999999
Customer code: I
Beginning reading value in gallons and tenths of gallon 0.4
Ending reading value in gallons and tenths of gallon 99999999.9
Gallons of water used: 99999999.5
Amount Billed$: 24500.00
➤ █
```

Project 2 Output for Customer Code (R)

Console Shell

```
Enter customer code (R, C, o
Enter beginning reading (between 0 and 999999999):999999990
Enter end reading (between 0 and 999999999):1230
Customer code: R
Beginning reading value in gallons and tenths of gallon 99999999.0
Ending reading value in gallons and tenths of gallon 123.0
Gallons of water used: 124.0
Amount Billed$: $5.06
➤ █
```


Project 2 Output for Customer Code (C)

Console Shell

```
Enter customer code (R, C, or I):C
Enter beginning reading (between 0 and 999999999):4
Enter end reading (between 0 and 999999999):9000000009
Customer code: C
Beginning reading value in gallons and tenths of gallon 0.4
Ending reading value in gallons and tenths of gallon 900000000.9
Gallons of water used: 900000000.5
Amount Billed$: $22500.00
> |
```

Project 2 Output for Invalid Customer Code

Console Shell

```
Enter customer code (R, C, or I):0
Invalid input(customer code)
> |
```

Project 2 Output for Out of Range Inputs

Console Shell

```
Enter customer code (R, C, or I):C
Enter beginning reading (between 0 and 999999999):9999999990
Enter end reading (between 0 and 999999999):54
Invalid input(beginning or ending reading value is out of the range)
> |
```