

Project2  
Water Conservation - Water Utility Company

**Objectives:**

1. To practice editing and running the code
2. To build up your skills to use programming for problem-solving
3. To focus on the design, implementation, and testing of a Python program with branches
4. To further participate in pair programming
5. To write a project report

**Introduction:**

Water conservation is one of the most pertinent issues in preserving our environment. We can do many things to ensure that we do not use more than we need to. Ten ways to save water at home are outlined on the following website:

<https://www.americanrivers.org/rivers/discover-your-river/top-10-ways-for-you-to-save-water-at-home/#:~:text=There%20are%20a%20few%20simple,shower%20head%20and%20faucet%20aerators.>

To play their part, Water Utility Companies charge different amounts per gallon depending on the consumption. The customers will save money if they save water – that is good additional motivation! A simplified method that a company would use to calculate the billed amount will be implemented in this project assignment.

**Problem Definition:**

1. The program will compute and display information for a utility company which supplies water to its customers. For a specified customer, the program will compute and display the amount of money which the customer will be billed for water usage during the current billing period.
2. The program will prompt the user to enter three values (in the following order):
  - a. The customer's code (a character) using a string 'Enter customer code (R, C, or I):'
  - b. The customer's beginning meter reading (a positive integer value), using a string 'Enter beginning reading (between 0 and 999999999):'
  - c. The customer's ending meter reading (a positive integer value) using a string 'Enter ending reading (between 0 and 999999999):'
3. Validate:
  - i. The program is **required** to check the customer's code immediately after the user provides the input value. Recognize **upper case only** letters for customer codes ("R", "C", and "I"). Any letter that is not defined as a code, input is considered an invalid input. If the code is not acceptable, print 'Invalid input (customer code)' and finish the program. Do not ask for other inputs.

- ii. If the customer's code is correct, the user should proceed to input both the start and end readings.

The program is **required** to check the user-supplied start and end reading immediately after **both** input values are read.

Validate that both values are within the range [0, 999999999].

Any value outside this range is considered invalid.

If any of the values for the readings are invalid, print ' Invalid input (beginning or ending reading value is out of the range)' and finish the program.

Do not ask the user to try again or perform any computation.

- 4. The program will compute the gallons of water used by the customer during the current billing period.

The meter is read by a representative of the utility company at the start and at the end of the billing period, and the readings are taken from a meter which has nine digits and records tenths of a gallon.

- 5. The program will compute the amount of money that the customer will be billed, based on the customer's code and water usage, using the following information.

Code 'r' (residential):

\$5.00 plus \$0.0005 per gallon used

Code 'c' (commercial):

\$1000.00 for 4 million gallons or less, and \$0.00025 for each additional gallon used

Code 'i' (industrial):

\$1000.00 if usage does not exceed 4 million gallons;

\$2000.00 if usage exceeds 4 million gallons but does not exceed 10 million gallons; and

\$2000.00 plus \$0.00025 for each additional gallon if usage exceeds 10 million gallons.

- 6. For each run, the program will display a summary with the following information:

- a. The customer's code
- b. The customer's beginning meter reading
- c. The customer's ending meter reading
- d. The gallons of water used by the customer
- e. The amount of money billed to the customer

- 7. validation and extra requirements:

- a. Recognize **upper case** letters for customer codes ("R", "C" and "I").

- b. Display all meter readings as **nine-digit numbers** (with leading zeroes).

**hint:** observe the output of this statement. Modify and use to match the requirements  
`print("{:0>4}".format(5))`

- c. The amount of money billed to a customer should be **displayed as a monetary** value. Meaning, it should be displayed with a dollar sign and two fractional digits (for example, \$125.00 or \$43.87).

- d. The program is **required** to check the customer's code and the user-supplied meter readings for errors.  
report these errors -**only, do not** ask the user to try again- invalid user inputs -Just

output an invalid message with zero as the value to be outputted at the end of the code  
-assuming we performed calculations-

- i. Any letter that is not defined as a code in the problem.
- ii. values outside the range 0 to 999999999

**Please Note:**

1. The meter's dial has **nine** digits and records **tenths of a gallon**.  
For example, assuming that the beginning reading was 444400003 and the ending reading was 444400135, then the customer used **13.2** gallons of water during the billing period.
2. Since the meter's dial only has nine digits, the reading at the end of the billing period **may be less than** the reading at the beginning of the billing period.  
For example, assuming that the beginning reading was 999999997 and the ending reading was 000000005, then the customer used 0.8 gallons of water during the billing period.
3. Please note that the user should always enter three items (code and two meter readings), even if the first meter reading is invalid.

**Suggestions to write any code:**

- **Solve the problem using pencil and paper first.** You cannot write a program until you have figured out how to solve the problem.
- It is fine if you discuss the paper & pencil solution with your friends, but you are to write **your own Python code**. Any similarities in coding both submissions will earn a zero.
- Use IDLE to create a new program.
- Write a simple version of the program -perhaps one which prompts the user for a customer code and displays it-.
- Run the program and track down any errors.
- Cycle through the steps to incrementally develop your program:
- Edit your program to add new capabilities.
- Run the program and fix any errors.
- Add comments explaining your steps.
- Do not use anything we did not learn yet.

Start with a small code...modify... modify until you reach your goal



Be ready to demonstrate your code in the lab. -you will be informed by your lab instructor-

Sample Output **Red-input**, **Blue output**

**Sample run1:**

Enter customer code (R, C, or I): **0**  
Invalid input (customer code)

Or for the invalid code you can output the following

Enter customer code (R, C, or I): **T**  
Beginning meter reading (between 0 and 999999999): **999999997**  
Ending meter reading (between 0 and 999999999): **000000005**  
Invalid Entry

Gallons of water used: 0.0  
Amount billed: \$0.00

**Sample run2:**

Enter customer code (R, C, or I): **R**  
Beginning meter reading (between 0 and 999999999): **444400003**  
Ending meter reading (between 0 and 999999999): **444400135**

Customer code: R  
Beginning meter reading: 444400003  
Ending meter reading: 444400135  
Gallons of water used: 13.2  
Amount billed: \$5.01

**Sample run3:**

Enter customer code (R, C, or I): **C**  
Beginning meter reading (between 0 and 999999999): **999999997**  
Ending meter reading (between 0 and 999999999): **000000005**

Customer code: C  
Beginning meter reading: 999999997  
Ending meter reading: 000000005  
Gallons of water used: 0.8  
Amount billed: \$1000.00  
Process finished with exit code 0

**Sample run4:**

Enter customer code (R, C, or I): **C**  
Beginning meter reading (between 0 and 999999999): **999999997**  
Ending meter reading (between 0 and 999999999): **000000005**