

[← Back to study plan](#)

Work on project. Stage 5/6: On a coffee loop

Project: [Coffee Machine](#)

Hard 29 minutes

3529 users solved this problem. Latest completion was about 10 hours ago.

Description

Just one action is not so interesting, is it? Let's improve the program so it can do multiple actions, one after another. It should repeatedly ask a user what they want to do. If the user types "buy", "fill" or "take", then the program should do exactly the same thing it did in the previous step. However, if the user wants to switch off the coffee machine, they should type "exit". The program should terminate on this command. Also, when the user types "remaining", the program should output all the resources that the coffee machine has.

4 / 4 Prerequisites

- ✓ List 14★ *** Stage 5
- ✓ While loop 8★ *** Stage 5
- ✓ For loop 9★ *** Stage 5
- ✓ Loop control statements 5★ *** Stage 5

Objectives

Write a program that will work endlessly to make coffee for all interested persons until the shutdown signal is given. Introduce two new options: "remaining" and "exit".

Do not forget that you can be out of resources for making coffee. If the coffee machine doesn't have enough resources to make coffee, the program should output a message that says it can't make a cup of coffee.

And the last improvement to the program at this step — if the user types "buy" to buy a cup of coffee and then changes his mind, they should be able to type "back" to return into the main cycle.

Example

Your coffee machine should have the the same initial resources as in the example (400 ml of water, 540 ml of milk, 120 g of coffee beans, 9 disposable cups, \$550 in cash).

The greater-than symbol followed by space (>) represents the user input. Notice that it's not the part of the input.

Example 1:

```
1 Write action (buy, fill, take, remaining, exit):
2 > remaining
3
4 The coffee machine has:
5 400 of water
6 540 of milk
7 120 of coffee beans
8 9 of disposable cups
9 $550 of money
10
11 Write action (buy, fill, take, remaining, exit):
12 > buy
13
14 What do you want to buy? 1 - espresso, 2 - latte, 3 - cappuccino, back - to main menu:
15 > 2
16 I have enough resources, making you a coffee!
17
18 Write action (buy, fill, take, remaining, exit):
19 > remaining
20
21 The coffee machine has:
22 50 of water
23 465 of milk
24 100 of coffee beans
25 8 of disposable cups
26 $557 of money
27
28 Write action (buy, fill, take, remaining, exit):
29 > buy
30
31 What do you want to buy? 1 - espresso, 2 - latte, 3 - cappuccino, back - to main menu:
32 > 2
33 Sorry, not enough water!
34
35 Write action (buy, fill, take, remaining, exit):
36 > fill
37
38 Write how many ml of water do you want to add:
```

```

39 > 1000
40 Write how many ml of milk do you want to add:
41 > 0
42 Write how many grams of coffee beans do you want to add:
43 > 0
44 Write how many disposable cups of coffee do you want to add:
45 > 0
46
47 Write action (buy, fill, take, remaining, exit):
48 > remaining
49
50 The coffee machine has:
51 1050 of water
52 465 of milk
53 100 of coffee beans
54 8 of disposable cups
55 $557 of money
56
57 Write action (buy, fill, take, remaining, exit):
58 > buy
59
60 What do you want to buy? 1 - espresso, 2 - latte, 3 - cappuccino, back - to main menu:
61 > 2
62 I have enough resources, making you a coffee!
63
64 Write action (buy, fill, take, remaining, exit):
65 > remaining
66
67 The coffee machine has:
68 700 of water
69 390 of milk
70 80 of coffee beans
71 7 of disposable cups
72 $564 of money
73
74 Write action (buy, fill, take, remaining, exit):
75 > take
76
77 I gave you $564
78
79 Write action (buy, fill, take, remaining, exit):
80 > remaining
81
82 The coffee machine has:
83 700 of water
84 390 of milk
85 80 of coffee beans
86 7 of disposable cups
87 0 of money
88
89 Write action (buy, fill, take, remaining, exit):
90 > exit

```

🔗 Write a program

[Code Editor](#) [IDE](#)

Python

```

1 # supply variables
2 stored_money = 550
3 stored_water = 400
4 stored_milk = 540
5 stored_beans = 120
6 stored_cups = 9
7
8
9 def prompt():
10     print('The coffee machine has:')
11     print(f'{stored_water} of water')
12     print(f'{stored_milk} of milk')
13     print(f'{stored_beans} of coffee beans')
14     print(f'{stored_cups} of disposable cups')
15     print(f'{stored_money} of money')
16     print()
17
18
19 def update_storage_info(money, water, milk, beans):
20     global stored_money
21     global stored_water
22     global stored_milk
23     global stored_beans
24     global stored_cups
25     stored_money = stored_money + money
26     stored_water = stored_water - water
27     stored_milk = stored_milk - milk
28     stored_beans = stored_beans - beans

```

```

29     stored_cups = stored_cups - 1
30
31
32 def refill(water, milk, beans, cups):
33     global stored_water
34     global stored_milk
35     global stored_beans
36     global stored_cups
37
38     stored_water = stored_water + water
39     stored_milk = stored_milk + milk
40     stored_beans = stored_beans + beans
41     stored_cups = stored_cups + cups
42
43
44 def fill():
45     water = int(input("Write how many ml of water do you want to add:\n"))
46     milk = int(input("Write how many ml of milk do you want to add:\n"))
47     beans = int(input("Write how many grams of coffee beans do you want to add:\n"))
48     cups = int(input("Write how many disposable cups of coffee do you want to add:\n"))
49
50     refill(water, milk, beans, cups)
51
52
53 def types_of_coffee():
54     coffee_flavor = int(input("What do you want to buy? 1"
55                               " - espresso, 2 - latte, 3 - cappuccino:\n"))
56
57     if coffee_flavor == 1:
58         update_storage_info(money=4, water=250, milk=0, beans=16)
59
60     elif coffee_flavor == 2:
61         update_storage_info(money=7, water=350, milk=75, beans=20)
62
63     elif coffee_flavor == 3:
64         update_storage_info(money=6, water=200, milk=100, beans=12)
65
66
67 def take():
68     global stored_money
69     cash = stored_money
70     stored_money = 0
71     print(f"I gave you ${cash}")
72
73
74 # first prompt
75 prompt()
76 # second prompt
77 answer = input("Write action (buy, fill, take):\n")
78
79 if answer == 'buy':
80     types_of_coffee()
81     print()
82     prompt()
83
84 elif answer == 'fill':
85     fill()
86     print()
87     prompt()
88
89 elif answer == 'take':
90     take()
91     print()
92     prompt()
93

```

[Run](#)
[Continue](#)
[Solutions \(403\)](#)

This content was created over 1 year ago and updated 5 days ago. [Share your feedback below in comments to help us improve it!](#)

[Show discussion \(806\)](#)