Cash

[tl;dr](https://docs.cs50.net/2019/x/psets/6/sentimental/cash/cash.html#tldr)

Implement a program that calculates the minimum number of coins required to give a user change.

$ python cash.py

Change owed: 0.41

4

[Specification](https://docs.cs50.net/2019/x/psets/6/sentimental/cash/cash.html#specification)

* Write, in a file called cash.py in ~/workspace/pset6/cash/, a program that first asks the user how much change is owed and then spits out the minimum number of coins with which said change can be made, exactly as you did in [Problem Set 1](https://lab.cs50.io/cs50/labs/2019/x/cash/), except that your program this time should be written (a) in Python and (b) in CS50 IDE.
* Use get\_float from the CS50 Library to get the user’s input and print to output your answer. Assume that the only coins available are quarters (25¢), dimes (10¢), nickels (5¢), and pennies (1¢).
  + We ask that you use get\_float so that you can handle dollars and cents, albeit sans dollar sign. In other words, if some customer is owed $9.75 (as in the case where a newspaper costs 25¢ but the customer pays with a $10 bill), assume that your program’s input will be 9.75 and not $9.75 or 975. However, if some customer is owed $9 exactly, assume that your program’s input will be 9.00 or just 9 but, again, not $9 or 900. Of course, by nature of floating-point values, your program will likely work with inputs like 9.0 and 9.000 as well; you need not worry about checking whether the user’s input is "formatted" like money should be.
* If the user fails to provide a non-negative value, your program should re-prompt the user for a valid amount again and again until the user complies.
* Incidentally, so that we can automate some tests of your code, we ask that your program’s last line of output be only the minimum number of coins possible: an integer followed by \n.

[Walkthrough](https://docs.cs50.net/2019/x/psets/6/sentimental/cash/cash.html#walkthrough)

**Note: In the past, this problem was called Greedy.**

https://youtu.be/f3iMmGry05Q

[Usage](https://docs.cs50.net/2019/x/psets/6/sentimental/cash/cash.html#usage)

Your program should behave per the example below. Assume that the underlined text is what some user has typed.

$ python cash.py

Change owed: 0.41

4

$ python cash.py

Change owed: -0.41

Change owed: -0.41

Change owed: foo

Change owed: 0.41

4

[Testing](https://docs.cs50.net/2019/x/psets/6/sentimental/cash/cash.html#testing)

[Correctness](https://docs.cs50.net/2019/x/psets/6/sentimental/cash/cash.html#correctness)

check50 cs50/problems/2019/x/sentimental/cash

[Style](https://docs.cs50.net/2019/x/psets/6/sentimental/cash/cash.html#style)

style50 cash.py

[Staff Solution](https://docs.cs50.net/2019/x/psets/6/sentimental/cash/cash.html#staff-solution)

~cs50/2019/x/pset6/cash

[How to Submit](https://docs.cs50.net/2019/x/psets/6/sentimental/cash/cash.html#how-to-submit)

Execute the below, logging in with your GitHub username and password when prompted. For security, you’ll see asterisks (\*) instead of the actual characters in your password.

submit50 cs50/problems/2019/x/sentimental/cash

You can then go to <https://cs50.me/cs50x> to view your current scores!