Lab 05: Object Oriented Programming

AUTHOR
Christian Osendorfer

Implement in Python

Unless otherwise stated, don't use any modules that implement a solution to the questions asked. Come up with your doctests and compare these with those from other students.

Vending Machine

In this question you'll create a vending machine that sells a single product and provides change when needed.

Create a class called `VendingMachine` that represents a vending machine for some product. A `VendingMachine` object returns strings describing its interactions. Make sure your output exactly matches the strings in the doctests including punctuation and spacing!

> **Note**
>
> Python's **f-strings** can be useful for string formatting. A quick example:
>
> ```
> >>> s1 = 'like'
> >>> s2 = 'python'
> >>> f'I {s1} {s2}!'
> 'I like python!'
> ```

Fill in the `VendingMachine` class, adding attributes and methods as appropriate, such that its behavior matches the following doctests:

```
class VendingMachine:
    """A vending machine that vends some product for some price.

    >>> v = VendingMachine('candy', 10)
    >>> v.vend()
    'Nothing left to vend. Please restock.'
    >>> v.add_funds(15)
    'Nothing left to vend. Please restock. Here is your $15.'
    >>> v.restock(2)
    'Current candy stock: 2'
    >>> v.vend()
    'Please add $10 more funds.'
    >>> v.add_funds(7)
    'Current balance: $7'
    >>> v.vend()
```

```
'Please add $3 more funds.'
>>> v.add_funds(5)
'Current balance: $12'
>>> v.vend()
'Here is your candy and $2 change.'
>>> v.add_funds(10)
'Current balance: $10'
>>> v.vend()
'Here is your candy.'
>>> v.add_funds(15)
'Nothing left to vend. Please restock. Here is your $15.'

>>> w = VendingMachine('soda', 2)
>>> w.restock(3)
'Current soda stock: 3'
>>> w.restock(3)
'Current soda stock: 6'
>>> w.add_funds(2)
'Current balance: $2'
>>> w.vend()
'Here is your soda.'
"""
# YOUR CODE HERE
```