



[CS 11 25.1] HOPE 3 – Fruits

Cheatsheet is available here: <https://oj.dcs.upd.edu.ph/cs11cheatsheet/>

Problem Statement

You want to sell fruits in your local neighborhood, and you want to write a program to help you manage your fruit store!

Task Details

You may import `Fruit`, `FruitKind`, and `Ripeness` from `oj`. These are implemented as follows:

```
from dataclasses import dataclass
from enum import auto, Enum

class FruitKind(Enum):
    ACCESSORY = auto()
    MULTIPLE = auto()
    SIMPLE = auto()
    AGGREGATE = auto()

class Ripeness(Enum):
    UNRIPE = auto()
    RIPE = auto()
    OVERRIPE = auto()

@dataclass
class Fruit:
    name: str
    kind: FruitKind
    ripeness: Ripeness
```

Copy

Submit solution
[CS 11 25.1]
HOPE 3

My submissions

✓ Points: 250 (partial)

⌚ Time limit: 12.0s

≡ Memory limit: 2G

➤ Problem type

▼ Allowed languages

py3

Your task is to implement a class named `FruitStore`. This function should have an initializer as well as three functions:

- The initializer should take in nothing.
- `add(name, kind)`: Adds a `Fruit` with name `name` and kind `kind` to your `FruitStore`. The newly added fruit is unripe.
- `pass_time()`: Makes time pass. All fruits that are unripe become ripe, all fruits that are ripe become overripe, and all fruits that are overripe rot, and you throw them away.
- `inventory(kind)`: Returns a list of the `names` of all `Fruit`s that you can sell. Should take in an **optional** argument `kind`. This list should be *sorted*. A fruit may be sold if it is not overripe.
 - If `kind` is not provided, returns all fruits you can sell.
 - Otherwise, returns all fruits of kind `kind` that you can sell.

Restrictions

Your source code must have at most 2,000 bytes.

Example Testing

```
store = FruitStore()

store.add("apple", FruitKind.ACCESSORY)
store.pass_time()
store.add("apple", FruitKind.ACCESSORY)
assert store.inventory() == ["apple", "apple"]
store.add("pineapple", FruitKind.MULTIPLE)
store.pass_time()
assert store.inventory(FruitKind.MULTIPLE) == ["pineapple"]
assert store.inventory() == ["apple", "pineapple"]
store.pass_time()
assert store.inventory() == []
```

Copy

Constraints

- The class `FruitStore` will be instantiated at most 50 times.
- The methods of the `FruitStore` class will be called at most 500 times.
- Each fruit name is a string of lowercase English letters with length at most 8.

Scoring

Note: New tests may be added and all submissions may be rejudged at a later time. (All future tests will satisfy the constraints.)

- You get 40 ❤ points if you solve all test cases where:
 - Every time `inventory` is called, `kind` is always provided.

- You get 40 ❤ points if you solve all test cases where:
 - Every time `inventory` is called, `kind` is never provided.

- You get 170 🟠 points if you solve all test cases.

FAQ Clarifications

Report an issue

No clarifications have been made at this time.