Solutions

Marking: **M1** a correct method, **A1** the correct answer, **B1** an independent mark.

Each answer shows the steps, then the **result**.

2.1

Car has an Engine; Student is a Person

Library has Books; Square is a Shape

Stack has a list (it should not inherit every list method)

2.2

`hi!`

`hi`

2.3

`AttributeError: 'DVD' object has no attribute 'title'`

`DVD.__init__` replaces `Item.__init__`, and never calls it

fix: `super().__init__(title)` at the start of `DVD.__init__`

2.4

`"B"`

lookup goes C, then B, then A, and stops at the first `f` it finds

3.1

(a) **6**, not 3

(b) `add_all` adds 3; then `Receipt.add_all` calls `self.add`, which is the subclass's `add`, adding 1 three more times

(c)

```
class CountingReceipt:
    def __init__(self):
        self._r = Receipt()
        self.count = 0
    def add(self, p):
        self.count += 1
        self._r.add(p)
    def add_all(self, ps):
        for p in ps:
            self.add(p)
```

holds a `Receipt` instead of inheriting; counts once per item in its own methods; passes the real work to the receipt

3.2

(a) $4 \times 4 = \mathbf{16}$

(b) callers of `Rectangle` expect width and height to change separately; `Square` changes that behaviour, so correct code for rectangles fails (Liskov substitution)

(c) do not make `Square` a subclass of a changeable `Rectangle`; e.g. both are immutable `Shape`s with `area()`, or a square is made by a `Rectangle.square(side)` factory

3.3

(a) DVD $3 \times$ £1.00 = £**3.00**; book $10 \times$ £0.20 = £**2.00**

(b)

```
@dataclass(frozen=True)
class LoanPolicy:
    days: int
    fine_per_day: float

class Item:
    def __init__(self, title, policy):
        self.title, self.policy = title, policy
    def fine(self, days_late):
        return days_late * self.policy.fine_per_day
```

a policy with the two numbers; the item holds a policy; `fine` passes the work to it

(c) subclasses: a new class, and maybe a deeper tree; policy: one new `LoanPolicy(2, 0.50)` object, no new code; recommend the policy design, because the types differ only in data, not behaviour

exercises.lol exercises.com.cn