

Solutions

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

2.1

- `def area(w, h):` with `return w * h`
- `print(area(3, 5))`

2.2

- (i) price or qty
- (ii) total
- (iii) RATE

2.3

- (i) **"a"**: keeps the earlier readings
- (ii) **"r"**: the file is only read
- (iii) **"w"**: the old contents should go

2.4

- `["Lee", "17", "B"]`, all strings

3.1 B

- write mode empties an existing file first

3.2 C

- `f` has no `return`, so it returns `None`

3.3 B

- modularity

3.4

(a) One answer:

```
def joined_before(filename, year):
    count = 0
    with open(filename, "r") as f:
        for line in f:
            name, joined = line.strip().split(",")
            if int(joined) < year:
                count = count + 1
    return count
```

- opens the file in read mode
- loops over the lines
- strips and splits each line
- casts the year and compares with <

- counts and returns the count after the loop

(b)

- Mei and Raj: **2**

(c)

- `with open("members.txt", "a") as f:`
- `f.write("Zoe,2026\n")`, with the newline

(d)

- `open` raises a `FileNotFoundError` and the program stops
- put the code in `try ... except FileNotFoundError` and print a message or return 0