

Step 7 — Refine with GenAI

Gen AI Name: Copilot

My Prompt:

Can you review my café checkout Python skeleton and suggest any improvements?

Response:

1. Code Organization & Modularity

Break globals out of the module scope and encapsulate state in a class or dataclass. This cleans up function signatures and makes it easy to spin up multiple rooms if needed.

Python

복사

```
from dataclasses import dataclass, field

@dataclass
class Classroom:
    projector_on: bool = False
    capacity: int = 30
    topic: str = ""
    attendance: set[str] = field(default_factory=set)
    temperatures: list[float] = field(default_factory=list)
```

Then methods can live on `Classroom` instead of touching globals.

Reflection:

I preceded this analysis by recording what I thought were proper global variables; namely, room, attendance, and temperatures. I replaced them with a dataclass called 'Classroom', which is a better architecture as it packs all related room state in an object. The suggestion from the Copilot was helpful and accepted as the program's enhanced structure, included the problem of the passing of several variables into function arguments, and made the program easier to maintain. The dataclass's user also improves the data structure, which facilitates clean and clear extensions, such as monitoring multiple rooms. I tested the enhanced version and was able to verify the output was working well as expected.