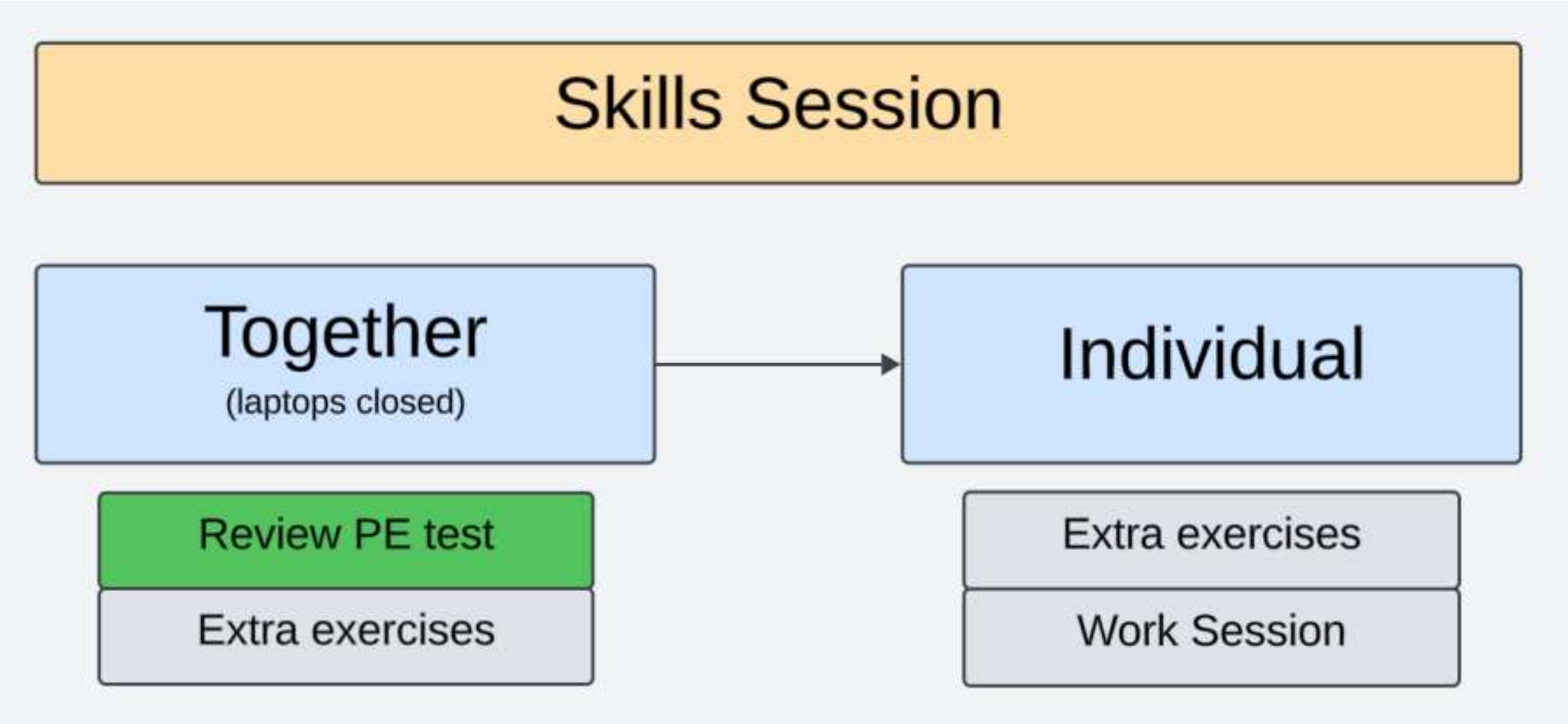
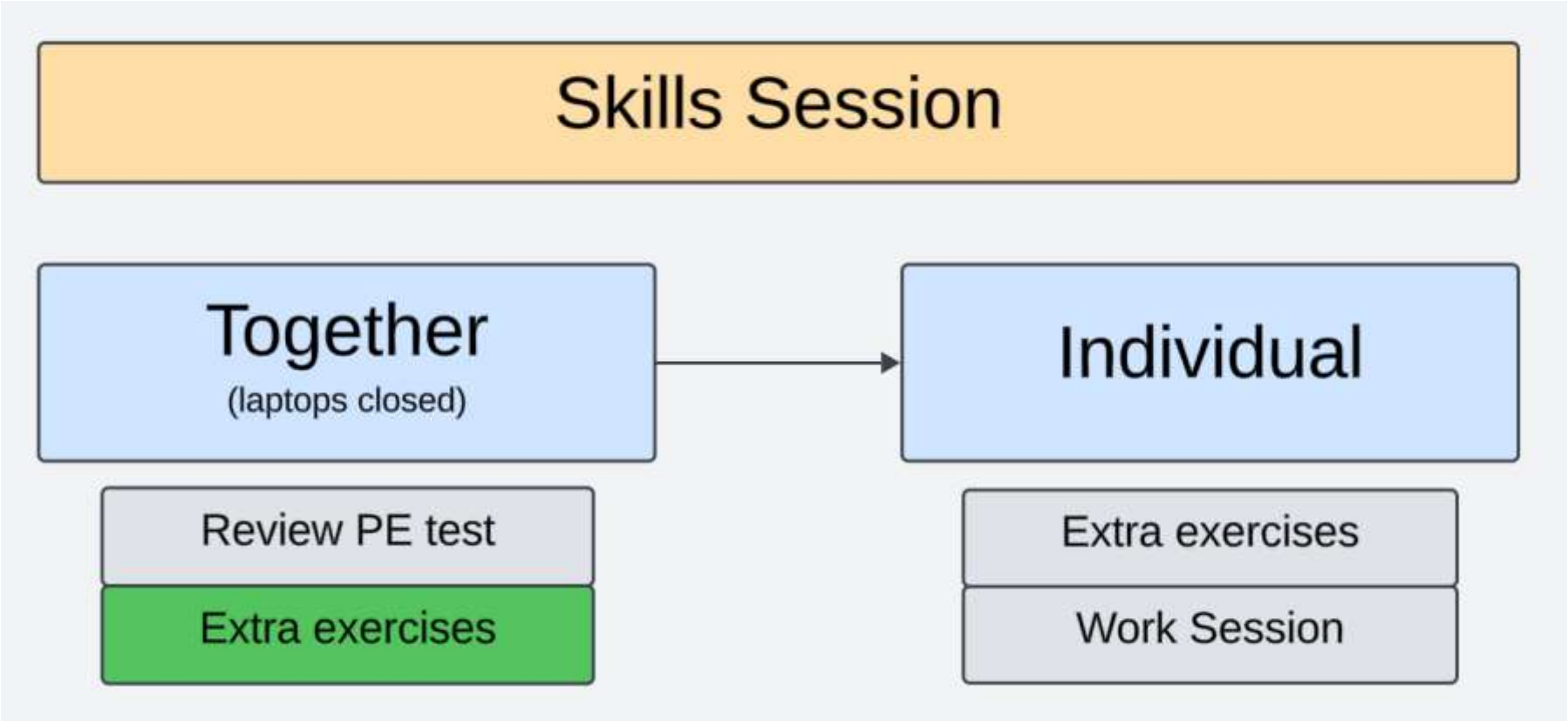
PROGRAMMING 1 - WEEK 6

Skills

Approach


Skills Session
Together
(laptops closed)
Individual
Review PE test
Extra exercises
Extra exercises
Work Session

Approach


Skills Session
Together
(laptops closed)
Individual
Review PE test
Extra exercises
Extra exercises
Work Session

Bookshop

We will implement some functionality for a **simplified** bookshop.

The bookshop holds 2 dictionaries:

- One with information about the books they sell
- One with information about the preferences of their customers

We are interested in adding information to the dictionaries, retreiving information, and combining information about books and customers.

Note that our implementation will not tackle case sensitivity.

The following weeks we will look into other concepts to tackle this implementation more efficient and user friendly!

Bookshop (1)

Given the dictionary **books**, with title as key, and a list of details as value.
The list of details consists of the author, the genre and the stock.

```
books = {
    "1984": ["George Orwell", "Fiction", 5],
    "Sapiens": ["Yuval Noah Harari", "Non-Fiction", 10],
    "To Kill a Mockingbird": ["Harper Lee", "Fiction", 8],
    "A Brief History of Time": ["Stephen Hawking", "Science", 7],
    "The Catcher in the Rye": ["J.D. Salinger", "Fiction", 6],
    "Becoming": ["Michelle Obama", "Biography", 12],
    "The Great Gatsby": ["F. Scott Fitzgerald", "Fiction", 4],
    "Homo Deus": ["Yuval Noah Harari", "Non-Fiction", 9],
    "The Subtle Art of Not Giving a F*ck": ["Mark Manson", "Self-Help", 15],
    "Educated": ["Tara Westover", "Biography", 10]
}
```

Bookshop (1)

Given the dictionary **books**, with title as key, and a list of details as value.
The list of details consists of the author, the genre and the stock.

Write the following functions:

- add_books (books , title , author , genre , stock)
  - If the title already exists in de dictionary, adjust the stock
  - No other input validation needed
- books_by_author (books , author)
  - Return a set with all books from the given author
- check_availability (books , title)
  - Return True if the title exists in the dictionary and the stock is above 0, False otherwise

Bookshop (2)

Given the dictionary **customers**, with name as key, and a set of genres as value.

```
customers = {
    "Alice": {"Fiction", "Non-Fiction"},
    "Bob": {"Fiction", "Science"},
    "Charlie": {"Biography", "Non-Fiction"},
    "Diana": {"Self-Help", "Fiction"},
    "Eve": {"Non-Fiction", "Science", "Biography"},
    "Frank": {"Fiction", "Biography"},
    "Grace": {"Self-Help", "Non-Fiction"}
}
```

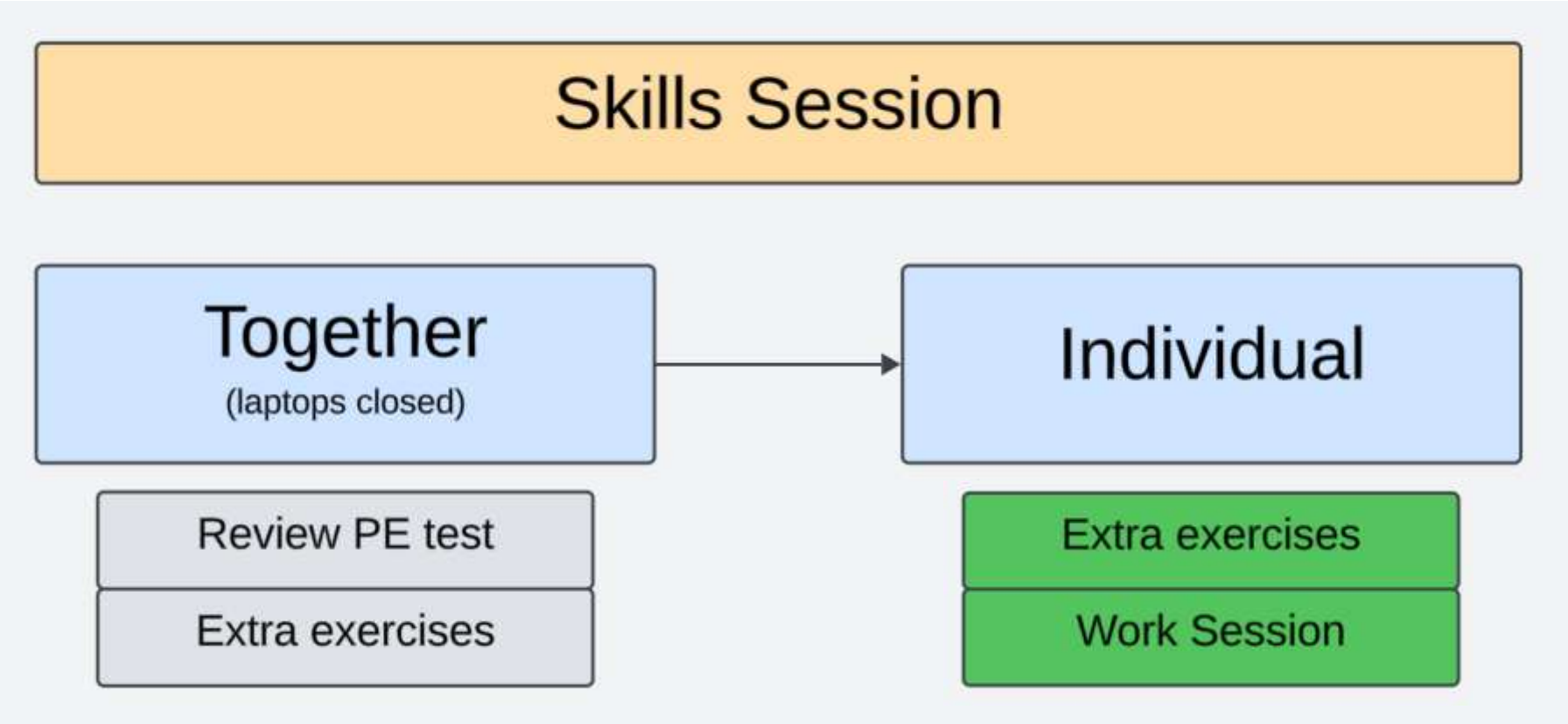
Bookshop (2)

Given the dictionary **customers**, with name as key, and a set of genres as value.

Write the following functions:

- add_genre (customers , name , genre)
  - If the customer doesn't exist in de dictionary, add the customer with the given genre
  - No other input validation needed
- all_genres (customers)
  - Return a set with all genres the customers are interested in
- recommend_books (books , customers , name)
  - Return a set with all book titles that correspond with the preferred genres of the given customer

Approach


Skills Session
Together
(laptops closed)
Individual
Review PE test
Extra exercises
Extra exercises
Work Session