

TX00FL42-3001

ASSOCIATION

MUATH OTHMAN



LET'S CHECK HOMEWORKS!



**LET'S DO
TOGETHER
EXERCISE 4!**



WHAT IS ASSOCIATION?



In object-oriented programming (OOP), association is a relationship between two or more objects that allows them to communicate and collaborate with each other. It defines how objects of different classes are connected and interact.

LET'S TAKE EXAMPLE









1. Introduction to Algorithms
2. Programming Python
3. Deep Learning

1. You Don't Know JS
2. Clean Code
3. Design Patterns

EXAMPLE: BOOK CLASS



```
class Book:  
    def __init__(self, title, author, ibsn):  
        self.title = title  
        self.author = author  
        self.ibsn = ibsn  
  
    def display(self):  
        print("Book Title: " + self.title + "\nAuthor: " + self.author + "\nISBN: " + str(self.ibsn))
```

EXAMPLE: LIBRARY CLASS

```
● ● ●

class Library:
    def __init__(self, name):
        self.name = name
        self.books = []

    def add_book(self, book):
        self.books.append(book)

    def list_books(self):
        print(f"Books in '{self.name}' Library:")
        for book in self.books:
            print("")
            book.display()
```

EXAMPLE: MAIN

```
● ● ●

from Book import Book
from Library import Library
from BorrowService import BorrowService

# Create books for Metropolia Myllypuro Library
book1 = Book("Introduction to Algorithms", "Thomas H. Cormen", 123456789)
book2 = Book("Programming Python", "Mark Lutz", 987654321)
book3 = Book("Deep Learning", "Ian Goodfellow", 112233445)

# Create books for Metropolia Karamalmi Library
book4 = Book("You Don't Know JS", "Kyle Simpson", 223344556)
book5 = Book("Clean Code", "Robert C. Martin", 334455667)
book6 = Book("Design Patterns", "Erich Gamma", 445566778)
```

EXAMPLE: MAIN



```
from Book import Book
from Library import Library
from BorrowService import BorrowService

# Create books for Metropolia Myllypuro Library
book1 = Book("Introduction to Algorithms", "Thomas H. Cormen", 123456789)
book2 = Book("Programming Python", "Mark Lutz", 987654321)
book3 = Book("Deep Learning", "Ian Goodfellow", 112233445)

# Create books for Metropolia Karamalmi Library
book4 = Book("You Don't Know JS", "Kyle Simpson", 223344556)
book5 = Book("Clean Code", "Robert C. Martin", 334455667)
book6 = Book("Design Patterns", "Erich Gamma", 445566778)

# Create libraries
myllypuro_library = Library("Metropolia Myllypuro Library")
karamalmi_library = Library("Metropolia Karamalmi Library")

# Add books to Metropolia Myllypuro Library
myllypuro_library.add_book(book1)
myllypuro_library.add_book(book2)
myllypuro_library.add_book(book3)
```

EXAMPLE: MAIN

```
● ● ●

from Book import Book
from Library import Library
from BorrowService import BorrowService

# Create books for Metropolia Myllypuro Library
book1 = Book("Introduction to Algorithms", "Thomas H. Cormen", 123456789)
book2 = Book("Programming Python", "Mark Lutz", 987654321)
book3 = Book("Deep Learning", "Ian Goodfellow", 112233445)

# Create books for Metropolia Karamalmi Library
book4 = Book("You Don't Know JS", "Kyle Simpson", 223344556)
book5 = Book("Clean Code", "Robert C. Martin", 334455667)
book6 = Book("Design Patterns", "Erich Gamma", 445566778)

# Create libraries
myllypuro_library = Library("Metropolia Myllypuro Library")
karamalmi_library = Library("Metropolia Karamalmi Library")

# Add books to Metropolia Myllypuro Library
myllypuro_library.add_book(book1)
myllypuro_library.add_book(book2)
myllypuro_library.add_book(book3)

# Add books to Metropolia Karamalmi Library
karamalmi_library.add_book(book4)
karamalmi_library.add_book(book5)
karamalmi_library.add_book(book6)
```

EXERCISE

CONFERENCE SYSTEM

Attendee class

- name (string)
 - attendee_id (string)
 - registered_sessions (list)
-
- register_for_session(session)
 - cancel_registration(session)
 - list_registered_sessions()

Session class

- session_title (string)
 - max_capacity (integer)
 - registered_attendees (list)
-
- add_attendee(attendee)
 - remove_attendee(attendee)
 - list_attendees()

If there is space for new attendee, add the Attendee to registered_attendees and prints a success message.

EXAMPLE: TEMPORARY ASSOCIATION

```
class BorrowService:
    def borrow_book(self, library, book):
        if book in library.books:
            library.remove_book(book)
        else:
            print("Book not found in the library")
        return library

    def return_book(self, library, book):
        library.add_book(book)
        return library
```