Abdullah Bilal

22108164

In [7]:

```python
class Book:
    def __init__(self, title, author, isbn, copies):
        self.title = title
        self.author = author
        self.isbn = isbn
        self.copies = copies

    def is_available(self):
        if self.copies > 0:
            return True
        else:
            return False

    def borrow(self):
        if self.is_available():
            self.copies -= 1
        else:
            print(f"The book '{self.title}' is not available to borrow.")

    def return_book(self):
        self.copies += 1


class Member:
    def __init__(self, name, member_id):
        self.name = name
        self.member_id = member_id
        self.borrowed_books = []

    def borrow_book(self, book):
        if book.is_available():
            self.borrowed_books.append(book.isbn)
            book.borrow()
            print(f"{self.name} borrowed '{book.title}'.")
        else:
            print(f"{self.name} couldn't borrow '{book.title}' because it's not available.")

    def return_book(self, book):
        if book.isbn in self.borrowed_books:
            self.borrowed_books.remove(book.isbn)
            book.return_book()
            print(f"{self.name} returned '{book.title}'.")
        else:
            print(f"{self.name} cannot return '{book.title}' because it wasn't borrowed.")

    def list_borrowed_books(self):
        if len(self.borrowed_books) > 0:
            print(f"{self.name} has borrowed these books:")
            for isbn in self.borrowed_books:
                print(f"- ISBN: {isbn}")
        else:
            print(f"{self.name} has no borrowed books.")


class Library:
    def __init__(self):
        self.books = {}
        self.members = {}

    def add_book(self, book):
        if book.isbn in self.books:
            print(f"Book '{book.title}' already exists. Adding {book.copies} more copies.")
            self.books[book.isbn].copies += book.copies
        else:
            print(f"Adding new book '{book.title}'.")
            self.books[book.isbn] = book

    def add_member(self, member):
        if member.member_id not in self.members:
            print(f"Adding member '{member.name}' with ID {member.member_id}.")
            self.members[member.member_id] = member
        else:
            print(f"Member with ID {member.member_id} already exists.")

    def find_book_by_title(self, title):
        result = []
        for book in self.books.values():
            if book.title.lower() == title.lower():
                result.append(book)
        return result

    def find_member_by_name(self, name):
        result = []
        for member in self.members.values():
            if member.name.lower() == name.lower():
                result.append(member)
        return result


lib = Library()

b1 = Book("Python Basics", "Alice", "12345", 3)
b2 = Book("Data Science Intro", "Bob", "67890", 2)
b3 = Book("Python Basics", "Alice", "12345", 2)

lib.add_book(b1)
lib.add_book(b2)
lib.add_book(b3)

m1 = Member("John", "M001")
m2 = Member("Doe", "M002")

lib.add_member(m1)
lib.add_member(m2)

print("\nBorrowing books:")
m1.borrow_book(b1)
m2.borrow_book(b2)
m1.borrow_book(b2)

print("\nReturning books:")
m1.return_book(b1)
m2.return_book(b2)
m2.return_book(b1)

print("\nListing borrowed books:")
m1.list_borrowed_books()
m2.list_borrowed_books()

print("\nSearching for books by title:")
found_books = lib.find_book_by_title("Python Basics")
for book in found_books:
    print(f"Found: {book.title}, Copies: {book.copies}")

print("\nSearching for members by name:")
found_members = lib.find_member_by_name("John")
for member in found_members:
    print(f"Found Member: {member.name}, ID: {member.member_id}")
```

```
Adding new book 'Python Basics'.
Adding new book 'Data Science Intro'.
Book 'Python Basics' already exists. Adding 2 more copies.
Adding member 'John' with ID M001.
Adding member 'Doe' with ID M002.

Borrowing books:
John borrowed 'Python Basics'.
Doe borrowed 'Data Science Intro'.
John borrowed 'Data Science Intro'.

Returning books:
John returned 'Python Basics'.
Doe returned 'Data Science Intro'.
Doe cannot return 'Python Basics' because it wasn't borrowed.

Listing borrowed books:
John has borrowed these books:
- ISBN: 67890
Doe has no borrowed books.

Searching for books by title:
Found: Python Basics, Copies: 5

Searching for members by name:
Found Member: John, ID: M001
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js