functions-programs

September 9, 2024

[1]:

```
def find_duplicates(lst):
    # Have we seen any elements before? If so, in which set?
    seen = set()
    # Which elements have duplicates? Start with an empty set.
    duplicates = set()

    # For each item in the list:
    for item in lst:
        # Is the item already in the 'seen' set?
        if item in seen:
            # If yes, is it a duplicate? Add to 'duplicates' set.
            duplicates.add(item)
        else:
            # If not, add the item to the 'seen' set for future checks.
            seen.add(item)

    # How can we return duplicates as a list?
    return list(duplicates)

# Example usage
my_list = [1, 2, 3, 4, 5, 2, 3, 6, 7, 4]
print("Duplicates:", find_duplicates(my_list))
```

```
Duplicates: [2, 3, 4]
```

[ ]:

```
def add_book(book_id, title, author):
    if book_id in books:
        print(f"Book ID {book_id} already exists.")
    else:
        books[book_id] = {"title": title, "author": author, "is_borrowed":␣
 ↪False}
        print(f"Added book: ID {book_id}, Title: {title}, Author: {author}")

def list_books():
    if not books:
        print("No books available.")
    else:
        for book_id, details in books.items():
            status = "Borrowed" if details["is_borrowed"] else "Available"
            print(f"ID: {book_id}, Title: {details['title']}, Author: {details['author']}, Status: {status}")

def add_user(user_id, name):
    if user_id in users:
        print(f"User ID {user_id} already exists.")
    else:
        users[user_id] = {"name": name, "borrowed_books": []}
        print(f"Added user: ID {user_id}, Name: {name}")

def borrow_book(user_id, book_id):
    if user_id not in users:
        print(f"User ID {user_id} does not exist.")
        return
    if book_id not in books:
        print(f"Book ID {book_id} does not exist.")
        return

    book = books[book_id]
    user = users[user_id]

    if book["is_borrowed"]:
        print(f"Book '{book['title']}' is already borrowed.")
    else:
        book["is_borrowed"] = True
        user["borrowed_books"].append(book_id)
        print(f"{user['name']} borrowed '{book['title']}'")

def return_book(user_id, book_id):
    if user_id not in users:
        print(f"User ID {user_id} does not exist.")
        return
    if book_id not in books:
        print(f"Book ID {book_id} does not exist.")
        return

    book = books[book_id]
    user = users[user_id]

    if book_id not in user["borrowed_books"]:
        print(f"{user['name']} did not borrow '{book['title']}'")
    else:
        book["is_borrowed"] = False
        user["borrowed_books"].remove(book_id)
        print(f"{user['name']} returned '{book['title']}'")
```

```
def main():
    while True:
        print("\nLibrary Management System")
        print("1. Add Book")
        print("2. List Books")
        print("3. Add User")
        print("4. Borrow Book")
        print("5. Return Book")
        print("6. Exit")

        choice = input("Enter your choice: ")

        if choice == '1':
            book_id = int(input("Enter book ID: "))
            title = input("Enter book title: ")
            author = input("Enter book author: ")
            add_book(book_id, title, author)
        elif choice == '2':
            list_books()
        elif choice == '3':
            user_id = int(input("Enter user ID: "))
            name = input("Enter user name: ")
            add_user(user_id, name)
        elif choice == '4':
            user_id = int(input("Enter user ID: "))
            book_id = int(input("Enter book ID: "))
            borrow_book(user_id, book_id)
        elif choice == '5':
            user_id = int(input("Enter user ID: "))
            book_id = int(input("Enter book ID: "))
            return_book(user_id, book_id)
        elif choice == '6':
            print("Exiting the system.")
            break
        else:
            print("Invalid choice, please try again.")

# Sample data
books = {
    1: {"title": "1984", "author": "George Orwell", "is_borrowed": False},
    2: {"title": "To Kill a Mockingbird", "author": "Harper Lee", "is_borrowed":
 ↪ False},
    3: {"title": "The Great Gatsby", "author": "F. Scott Fitzgerald",␣
 ↪"is_borrowed": False},
}

users = {
```

```python
    101: {"name": "Alice", "borrowed_books": []},
    102: {"name": "Bob", "borrowed_books": []},
}

if __name__ == "__main__":
    main()
```