CSE 1321L: Programming and Problem Solving I Lab

Assignment 7 – 100 points

Object Oriented Programming and Classes

What students will learn:

1) Designing classes including attributes, constructors and methods

2) Creating objects of those class types

Overview: The programs you've written up to this point have started at "main" and basically executed from "top to bottom". In the last assignment, functions/methods were introduced, where you learned that the execution of code could jump from main, to a function, then back to main. For this assignment, we're going to practice OOP, or Object-Oriented Programming. OOP is a completely different way to think about programming. The concept originated in the 1960's, but it wasn't widely adopted until the mid 1980's. So, what is it? At its core is the concept of a "class" which is simply a grouping of variables and methods; the behavior of those methods usually change based on the state (also known as values) of the variables.

Assignment 7A: Library System with Borrow Limits

Requirements:

1. Design a Library System using two classes: Book and Member.
2. The system should allow members to borrow and return books according to specific rules.
3. Book Class:
    - Attributes (define in __init__ method):
        - title (string): The title of the book.
        - author (string): The author of the book.
        - is_borrowed (boolean): Tracks whether the book is currently borrowed.
    - Methods:
        - borrow(): Marks the book as borrowed if it isn't already.
            - Returns "You borrowed the book" if successful, or "Book already borrowed" if it is.
        - return_book(): Marks the book as returned if it was borrowed.

        - Returns "You returned the book" if successful, or "Book is not borrowed" if it wasn't.

2. Member Class:

    - Attributes (define in __init__ method):

        - name (string): The name of the member.

        - borrowed_books (list): A list of Book objects currently borrowed by the member.

    - Methods:

        - borrow_book(book): Allows the member to borrow a book according to these rules:

            - Members can borrow a maximum of 3 books. If they try to borrow more, return "Borrow limit reached".

            - If the member has already borrowed the book, return "Book already borrowed by this member".

            - If another member has borrowed the book, return "Book already borrowed by someone else".

            - Otherwise, add the book to borrowed_books and return "Book borrowed successfully".

        - return_book(book): Allows the member to return a book according to these rules:

            - If the member didn't borrow the book, return "Book not borrowed by this member".

            - If they have, mark the book as returned, remove it from borrowed_books, and return "Book returned successfully".

Testing the Code

- Define a Book: Create a single book with a name of your choice and test its borrowing and returning functionalities.

- Define a Member: Create one member to test basic borrowing limits, borrowing the same book, and returning it.

Sample Input and Output

```
# Define a Book and Member for testing

book1 = Book("1984", "George Orwell")

member1 = Member("Alice")
```

```
# Scenario 1: Alice borrows the book "1984"
print(member1.borrow_book(book1))
# Output: "Book borrowed successfully"

# Scenario 2: Alice tries to borrow "1984" again
print(member1.borrow_book(book1))
# Output: "Book already borrowed by this member"

# Scenario 3: Alice returns "1984"
print(member1.return_book(book1))
# Output: "Book returned successfully"

# Scenario 4: Alice tries to return "1984" again (after already returning it)
print(member1.return_book(book1))
# Output: "Book not borrowed by this member"

# Scenario 5: Another member, Bob, tries to borrow "1984" after Alice has returned it
member2 = Member("Bob")
print(member2.borrow_book(book1))
# Output: "Book borrowed successfully"

# Scenario 6: Alice tries to borrow more than the limit of 3 books
#Define more books
book2 = Book("To Kill a Mockingbird", "Harper Lee")
book3 = Book("The Catcher in the Rye", "J.D. Salinger")
book4 = Book("Pride and Prejudice", "Jane Austen")
```

```
# Alice borrows 3 books
print(member1.borrow_book(book2))
# Output: "Book borrowed successfully"
print(member1.borrow_book(book3))
# Output: "Book borrowed successfully"
print(member1.borrow_book(book4))
# Output: "Book borrowed successfully"

# Alice tries to borrow a fourth book
print(member1.borrow_book(book5))
# Output: "Borrow limit reached"
```