IT5016 Week 6 Part 3 Lab

Building a Movie Rental System with Python OOP

Objective:

Develop a Python-based movie rental system using Object-Oriented Programming (OOP). This system will include classes for managing movies, customers, and a rental store. You will also create a text-based menu to interact with the system.

Requirements:

1. **Define the Classes:**
    - **Movie Class:**
        - **Attributes:**
            - title: The title of the movie.
            - genre: The genre of the movie.
            - year: The release year of the movie.
            - available: A boolean indicating whether the movie is available for rent.
        - **Methods:**
            - mark_as_rented(): Marks the movie as rented.
            - mark_as_available(): Marks the movie as available.
            - __str__(): Provides a string representation of the movie's details.
    - **Customer Class:**
        - **Attributes:**
            - name: The name of the customer.
            - rented_movies: A list to keep track of movies rented by the customer.
        - **Methods:**
            - rent_movie(movie): Allows the customer to rent a movie if it is available.
            - return_movie(movie): Allows the customer to return a rented movie.

        - list_rented_movies(): Displays a list of movies currently rented by the customer.
    - **RentalStore Class:**
        - **Attributes:**
            - movies: A list to store available movies in the rental store.
        - **Methods:**
            - add_movie(movie): Adds a new movie to the store's collection.
            - list_movies(): Displays all available movies in the store.
            - find_movie(title): Searches for a movie by its title.
2. **Create a Menu System:**
    - Implement a text-based menu to allow users to interact with the rental system. The menu should provide the following options:
        - List available movies.
        - Rent a movie.
        - Return a movie.
        - List rented movies for a specific customer.
        - Add a new movie to the store.
        - Exit the program.
3. **Implement the Menu System:**
    - Display the menu to the user and prompt them to select an option.
    - Based on the user's choice, execute the corresponding action by calling the appropriate methods on the RentalStore and Customer objects.
    - Ensure that the menu allows users to handle errors gracefully, such as entering invalid choices or trying to rent a non-existent movie.

Steps to Complete:

1. **Design and Implement Classes:**
    - Write the Movie, Customer, and RentalStore classes according to the specifications.
2. **Implement the Menu:**

- Write functions to display the menu and handle user input.
- Implement a loop to keep the menu active until the user chooses to exit.

3. **Testing:**
   - Test each functionality to ensure it works as expected. This includes adding movies, renting and returning movies, listing available and rented movies, and handling invalid inputs.

01_movie.py

```python
class Movie:
    def __init__(self, title, genre, year):
        self.title = title
        self.genre = genre
        self.year = year
        self.available = True

    def mark_as_rented(self):
        self.available = False

    def mark_as_available(self):
        self.available = True

    def __str__(self):
        return f"{self.title} ({self.year}) - {self.genre} - {'Available' if self.available else 'Rented'}"
```

```python

class Customer:
    def __init__(self, name):
        self.name = name
        self.rented_movies = []

    def rent_movie(self, movie):
        if movie.available:
            movie.mark_as_rented()
            self.rented_movies.append(movie)
            print(f"{self.name} rented {movie.title}")
        else:
            print(f"{movie.title} is not available")

    def return_movie(self, movie):
        if movie in self.rented_movies:
            movie.mark_as_available()
            self.rented_movies.remove(movie)
            print(f"{self.name} returned {movie.title}")
        else:
            print(f"{self.name} did not rent {movie.title}")

    def list_rented_movies(self):
        if self.rented_movies:
            print(f"{self.name}'s Rented Movies:")
            for movie in self.rented_movies:
                print(movie)
        else:
            print(f"{self.name} has no rented movies.")

```

```python

class RentalStore:
    def __init__(self):
        self.movies = []

    def add_movie(self, movie):
        self.movies.append(movie)

    def list_movies(self):
        if self.movies:
            print("Available Movies:")
            for movie in self.movies:
                print(movie)
        else:
            print("No movies available.")

    def find_movie(self, title):
        for movie in self.movies:
            if movie.title.lower() == title.lower():
                return movie
        return None

def menu():
    print("\nMovie Rental System Menu")
    print("1. List Available Movies")
    print("2. Rent a Movie")
    print("3. Return a Movie")
    print("4. List Rented Movies")
    print("5. Add a Movie to Store")
    print("6. Exit")

```

```python
def main():
    # Initialize the store and movies
    store = RentalStore()
    store.add_movie(Movie("Inception", "Sci-Fi", 2010))
    store.add_movie(Movie("The Matrix", "Action", 1999))
    store.add_movie(Movie("The Godfather", "Crime", 1972))

    # Initialize customers
    customers = {
        "Alice": Customer("Alice"),
        "Bob": Customer("Bob")
    }

    while True:
        menu()
        choice = input("Enter your choice: ").strip()

        if choice == "1":
            store.list_movies()
        elif choice == "2":
            customer_name = input("Enter your name: ").strip()
            movie_title = input("Enter movie title to rent: ").strip()
            customer = customers.get(customer_name)
            movie = store.find_movie(movie_title)
            if customer and movie:
                customer.rent_movie(movie)
            elif not customer:
                print("Customer not found.")
            elif not movie:
                print("Movie not found.")
```

```
        elif choice == "3":
            customer_name = input("Enter your name: ").strip()
            movie_title = input("Enter movie title to return: ").strip()
            customer = customers.get(customer_name)
            movie = store.find_movie(movie_title)
            if customer and movie:
                customer.return_movie(movie)
            elif not customer:
                print("Customer not found.")
            elif not movie:
                print("Movie not found.")
        elif choice == "4":
            customer_name = input("Enter your name: ").strip()
            customer = customers.get(customer_name)
            if customer:
                customer.list_rented_movies()
            else:
                print("Customer not found.")
        elif choice == "5":
            title = input("Enter movie title: ").strip()
            genre = input("Enter movie genre: ").strip()
            year = int(input("Enter movie year: ").strip())
            store.add_movie(Movie(title, genre, year))
            print(f"Movie '{title}' added to the store.")
        elif choice == "6":
            print("Exiting...")
            break
        else:
            print("Invalid choice, please try again.")

if __name__ == "__main__":
    main()

```