

Assessment Problem Statement

Title: *Optimizing a Video Streaming Platform using Flyweight and Proxy Patterns*

Context

You are working as a software engineer for **StreamFlix**, a global video streaming platform similar to Netflix.

The platform has millions of users watching thousands of movies and TV shows every day. Each movie contains **metadata** (title, genre, language, director, actors, etc.) and **video files** that must be accessed efficiently.

Challenges

1. Each movie object currently stores all its metadata separately, leading to **huge memory usage** because the same movie can be streamed by millions of users.
2. Some video content is hosted on **external servers**, and fetching it repeatedly for each user causes **network delay and cost**.

Your Task

Design a system that:

1. **Uses the Flyweight Pattern** to minimize memory consumption by **sharing common movie metadata** among multiple streaming sessions.
2. **Uses the Proxy Pattern** to handle **video content access**, adding a **cache** to avoid redundant remote fetches.

Requirements

- Each user's streaming session should contain:
 - A reference to a shared MovieFlyweight object (metadata)
 - A VideoProxy object that fetches or reuses video content
- When the same movie is watched by multiple users, the shared movie data should not be duplicated.
- When a user watches a movie that was recently accessed, the proxy should serve it from cache rather than re-downloading it.

Deliverables

1. UML-style class structure (or class explanation)
2. Python implementation
3. Explanation of how Flyweight and Proxy optimize performance