

# DESIGN PROBLEM

The local Blockbuster wants to update their inventory system and your project team has been hired. As they sell movies, as well as rent them, you are assigned to create the module that will handle the pricing for movies. When a movie is first released, it is sold for the full price. After 3 months, there is less interest in the movie, so to encourage sales the price is dropped to 75% of full price. After another 3 months (i.e. six months after release) the movie's selling price drops to 50%, and after a year it drops to 25% to clear out any leftovers.

**Write a small C++ program that uses a design pattern presented in class to manage the price of movies.**

A class to represent a movie has been provided (`Movie.h`), along with a `main()`. Implement the `getPrice()` method, along with whatever other classes are needed. So that your module is resource-efficient, there can only be one of each class at runtime (in the actual running system, there would be multiple instances of `Movie`, but in the provided program there will only be the one that is created in the `main()`).

The output of the program should look like:

[illegible]