

```
// -----movie.h-----
// Adam Ali / Omar Aguirre CSS 343 A
// Created: 05/21/17
// Modified: 05/23/17
// -----
// Describes the abstract ADT Movie such that any genre/category of movie can
// be held in the store. All movies have a director, title, and release year.
// Derived classes of Movie add whatever attributes pertain to their genre,
// but all are equipped with operations to retrieve their attributes, display
// details, and make comparisons. All derivatives will have a const string
// to record their genre, which should be identical to the class name.
// -----
// Functionality includes:
//      - create a Movie item
//      - copy an existing Movie item
//      - destruct a Movie item
//      - retrieve attributes
//      - display details
//      - comparison operators (==, <)

#pragma once

#include <string>
#include "media.h"
using namespace std;

class Movie : public Media {
public:
    // -----Movie-----
    // Movie: creates a specified Movie.
    // preconditions: strings are nonempty, year >= 0.
    // postconditions: a Movie is created with specified fields.
    // -----
    Movie(string, string, string, int);

    // -----Movie-----
    // Movie: creates a copy of the Movie.
    // preconditions: none.
    // postconditions: a copy of the Movie is created.
    // -----
    Movie(const Movie&);

    // -----~Movie-----
    // ~Movie: deletes all assoc. memory. since variables for this class are
    //          statically alloc'd, this destructor enables an appropriate
    //          order for constructor calls for child classes.
    // preconditions: none.
    // postconditions: all assoc. memory de-alloc'd.
    // -----
    virtual ~Movie();

    // -----getDirector-----
    // getDirector: obtains the director.
    // preconditions: none.
    // postconditions: director is returned.
    // -----
    string getDirector() const;

    // -----getTitle-----
    // getTitle: obtains the title.
```

```
// preconditions: none.
// postconditions: title is returned.
// -----
string getTitle() const;

// -----getYear-----
// getYear: obtains the year.
// preconditions: none.
// postconditions: year is returned.
// -----
int getYear() const;

// -----display-----
// display: output Movie details to console. implementation left to
//         children.
// preconditions: none.
// postconditions: Movie remains unchanged.
// -----
virtual void display() const = 0;

// -----operator==-----
// operator==: compares both Media objects to check if they are equal.
//             implemented by child classes.
// preconditions: none.
// postconditions: true if identical, otherwise false.
// -----
virtual bool operator==(const Media&) const = 0;

// -----operator<-----
// operator<: compares this Media object to check if it precedes the
//            other. implemented by child classes.
// preconditions: none.
// postconditions: true if preceding, otherwise false.
// -----
virtual bool operator<(const Media&) const = 0;

private:
    string director;
    string title;           // film title
    int year;               // year of release

};
```