```
// ----------------------------------Comedy.h-------------------------------
// Adam Ali / Omar Aguirre CSS 343 A
// Created: 05/21/17
// Modified: 05/23/17
// -------------------------------------------------------------------------
// Describes the ADT Comedy such that a particular Comedy genre can not only
// maintain a record of director, title and year (as all Movie genres do), but
// to also record any other details that may be included later (now, none).
// The const string is the main differentiator between the other derivatives
// of Movie.

// Comedy is a child class derived from Movie.
// -------------------------------------------------------------------------
// Functionality includes:
//              - create a Comedy item
//      - copy an existing Comedy item
//              - destruct a Comedy item
//              - retrieve attributes
//              - display details
//              - comparison operators (==, <)

#pragma once

#include <string>
#include "movie.h"

class Comedy : public Movie {
public:
        // --------------------------------Comedy-----------------------------
        // Comedy: creates a Comedy item. No additional attributes.
        // preconditions: none.
        // postconditions: a Comedy item is created.
        // -------------------------------------------------------------------
        Comedy();


        // --------------------------------Comedy-----------------------------
        // Comedy: copies the Comedy item.
        // preconditions: none.
        // postconditions: a copy of the Comedy item is created.
        // -------------------------------------------------------------------
        Comedy(const Comedy&);


        // -------------------------------~Comedy-----------------------------
        // Comedy: frees all (static) alloc'd memory by engaging the right
        //                 sequence of destructors -- from child to parent.
        // preconditions: none.
        // postconditions: all (static) memory de-alloc'd.
        // -------------------------------------------------------------------
        virtual ~Comedy();


        // -------------------------------display-----------------------------
        // display: outputs Comedy details.
        // preconditions: none.
        // postconditions: Comedy details output to console.
        //                                 Comedy remains unchanged.
        // -------------------------------------------------------------------
        virtual void display() const;


        // ----------------------------operator==-----------------------------
        // operator==: determines if both items are identical, based on
```

```
        //                 attributes common to all Movie genres.
        // preconditions: none.
        // postconditions: true if identical, otherwise false.
        // ------------------------------------------------------------------
        virtual bool operator==(const Media&) const;


        // ----------------------------operator<---------------------------
        // operator<: compares this Comedy object to check if it precedes the
        //                        other.
        // preconditions: none.
        // postconditions: true if preceding, otherwise false.
        // ------------------------------------------------------------------
        virtual bool operator<(const Media&) const {
                //if titles are equal then compare directors
                //if directors are equal then check for year published
                // if this.title < Media.title
                        // return true
                // else if this.director < Media.director
                        // return true
                // else if this.year < Media.year
                        // return true

                // return false //None of the previous conditions were met
        }

private:
        const string CATEGORY = "COMEDY";
};
```