**Movie, Comedy, Drama, and Classic Class Descriptions**

class movie {

public:

virtual Movie( ) = 0; // default constructor

// constructor with parameters for each data member

virtual Movie( int& stock, string& director, string& title, int& yearOfRelease, string& movieCode) = 0;

vitual Movie( Movie& ) = 0; // copy constructor

virtual ~Movie( ) = 0; // destructor

// create is a factory that creates the correct movie type based on the parameter and returns a new object

static Movie\* create(string type);

virtual int getStock( ) = 0; // returns the available stock of the movie

virtual void setStock( int& ) = 0; // method to set the stock of the movie

virtual string getMovieType( ) = 0; // returns the type of the movie

virtual string getDirector( ) = 0; // returns the director of the movie

virtual void setDirector( string& ) = 0; // method to set the director of the movie

virtual string getTitle( ) = 0; // returns the title of the movie

virtual void setTitle( string& ) = 0; // method to set the title of the movie

virtual int getYearOfRelease( ); // returns the year of release of the movie

virtual void setYearOfRelease( int& ); // method to set the year of release of the movie

virtual bool operator==( Movie& ) = 0; // compares this movie and other movie for equality

virtual bool operator<( Movie& ) = 0; // tests to see if this movie is less than other movie

virtual bool operator>( Movie& ) = 0; // tests to see if this movie is greater than other movie

private:

string movieType; // stores the type of the movie. F for comedy. D for drama. C for comedy.

int stock; // this will represent the available stock of the movie

string director; // stores the name of the director of the movie

string title; // stores the title of the movie

int yearOfRelease; // stores the year of release of the movie

};

class comedy {

public:

// operator== pseudocode. Items are sorted by title and then year of release

operator== : bool

if this movie title and other movie title are equal

if this movie year of release and other movie year of release are equal

return true

return false

return false

// operator< pseudocode. Items are sorted by title and then year of release

operator< : bool

if this movie title and other movie title are equal

if this movie year of release is less than other movie’s year of release

return true

return false

if this movie title is less than other movie title

return true

return false

// operator> pseudocode. Items are sorted by title and then year of release

operator> : bool

if this movie title and other movie title are equal

if this movie year of release is less than other movie’s year of release

return true

return false

if this movie title is greater than other movie title

return true

return false

};

class drama {

public:

// operator== pseudocode. Items are sorted by director and then title

operator== : bool

if this movie director and other movie director are equal

if this movie title and other movie title are equal

return true

return false

return false

// operator< pseudocode. Items are sorted by director and then title

operator< : bool

if this movie director and other movie director are equal

if this movie title is less than other movie’s title

return true

return false

if this movie director is less than other movie director

return true

return false

// operator> pseudocode. Items are sorted by director and then title

operator> : bool

if this movie director and other movie director are equal

if this movie title is greater than other movie’s title

return true

return false

if this movie director is greater than other movie director

return true

return false

};

class classics {

public:

string getMajorActorName( ); // returns the name of the major actor of the movie

void setMajorActorName( string& ); // method to set the name of the major actor of the movie

int getMonthOfRelease( ); // returns the month of release of the movie

void setMonthOfRelease( int& ); // method to set the month of release of the movie

// operator== pseudocode. Items are sorted by release date and then major actor

operator== : bool

if this movie month and year of release and other movie month and year of release are equal

if this movie major actor and other movie major actor are equal

return true

return false

return false

// operator< pseudocode. Items are sorted by release date and then major actor

operator< : bool

if this movie month and year of release and other movie month and year of release are equal

if this movie major actor is less than other movie’s major actor

return true

return false

if this movie month and year of release is less than other movie month and year of release

return true

return false

// operator> pseudocode. Items are sorted by release date and then major actor

operator> : bool

if this movie month and year of release and other movie month and year of release are equal

if this movie major actor is greater than other movie’s major actor

return true

return false

if this movie month and year of release is greater than other movie month and year of release

return true

return false

private:

string majorActorName; // stores the name of the major actor of the movie

int monthOfRelease; // stores the month of release of the movie

};