CSCE 240 – Programming Assignment Five

Due: 11:59pm on Wednesday, April 10th

Project Purpose

Implement the two classes described below. Both classes must be added to the csce240_programming_assignment_5 namespace.

NOTE: The only standard library includes allowed for this project are: iostream and string. You must properly manage memory using pointers.

SongRecording

private data members:

- a string for the song title
- a pointer to dynamically allocated memory holding the artist name(s) (strings)
- an integer hold the number of artists (must always be greater than or equal to 1)
- an integer to hold the length of the track in seconds (must always be non-negative)

public member functions:

- A constructor that can take the title, primary artist, track length, and number of artists as its four parameters with default arguments of “untitled”, “unknown”, 0, and 1, respectively

- A copy constructor

- An overloaded assignment operator

- A destructor

- GetTitle() - Returns the title

- SetTitle(string) – Sets the title to any string that is at least 1 character long

- GetNumArtists() – Returns the number of artists

- SetNumArtists(int) – Sets the number of artists to any positive integer. Manages memory for the string pointer data member.

- SetArtist(string, int) – Sets the name of the artist if the string is at least one character in length, and the integer argument is valid (between 1 and the number of artists, inclusive). The int parameter should have a default argument of 1.

- GetArtist(int) – This function should have a default argument of 1. It returns the name of the specified artist if the integer argument is between 1 and the number of artists, inclusive. If the integer argument is out of range, the function should return the string “out of bounds”

- GetTrackLength() – Returns the track length

- SetTrackLength(int) - Sets the value of the track length data member to the functions argument as long as the argument is non-negative. If the argument is negative the track length data member should remain unchanged.

Initial testing of the SongRecording class

Include your SongRecording.h and SongRecording.cc files in the same directory as all of the provided test files and makefile. Type the following commands for initial tests of the data members and functionality.

```
make testTitle
make testTrackLength
make testArtist
make testSoundRecordingConstructors1
make testSoundRecordingCopyConstructor
make testSoundRecordingAssignment
```

You are encouraged to create additional, more rigorous tests!

StreamingTrack – This class must be derived via public inheritance from your SongRecording class

private data members:

- an integer holding the number of times the track has been streamed
- a pointer to dynamically allocated memory holding the track's genre(s) (strings)
- an integer hold the number of genres (must be non-negative)

public member functions:

- A constructor that can take the title, primary artist, track length, number of artists, primary genre, and number of streams as its six parameters with default arguments of "untitled", "unknown", 0, 1, "pop", and 0 respectively

- A copy constructor

- A constructor that takes a constant reference to a SongRecording, a primary genre (with a default argument of "pop"), and a number streams (with a default argument of 0) as its three parameters

- An overloaded assignment operator

- A destructor

- GetStreams() - Returns the number of streams

- SetStreams(int) - Sets the number of streams to the argument as long as the argument is non-negative. If the argument is negative, the data member should remain unchanged.

- AddStreams(int) - Increases the number of streams by the argument as long as the argument is non-negative. If the argument is negative, the data member should remain unchanged.

- GetNumGenres() - Returns the number of genres

- GetGenre(int) - This function should have a default argument of 1. It returns the name of the specified genre if the integer argument is between 1 and the number of genres, inclusive. If the integer argument is out of range, the function should return the string “out of bounds”

- IsGenre(string) - Returns true if the string argument is one of the StreamingTrack’s genres, and false otherwise

- AddGenre(string) - Adds the string argument to the StreamingTrack’s genres (updating the pointer data member and the number of genres data member) if the argument is not already in the list of genres. If the argument is already one of the track’s genres, the data members should remain unchanged

- RemoveGenre(string) - Removes the string argument from the StreamingTrack’s genres (updating the pointer data member and the number of genres data member) if the argument is in the list of genres. If the argument is not one of the track’s genres, the data members should remain unchanged

Initial testing of the StreamingTrack class

Include your SongRecording.h, SongRecording.cc, StreamingTrack.h, and StreamingTrack.cc files in the same directory as all of the provided test files and makefile. Type the following commands for initial tests of the data members and functionality.

```
make testStreams
make testAddGenre
make testRemoveAndAddGenre
make testIsGenre
make testStreamingTrackConstructors1
make testStreamingTrackCopyConstructor
make testStreamingTrackAssignment
make testStreamingTrackBaseToDerivedConstructor
```

You are encouraged to create additional, more rigorous tests!

Additional Specifications

- The definition of the SongRecording class must be written in a file named SongRecording.h
- The implementation of the member functions for the SongRecording class must be written in a file named SongRecording.cc
- The definition of the StreamingTrack class must be written in a file named StreamingTrack.h
- The implementation of the member functions for the StreamingTrack class must be written in a file named StreamingTrack.cc
- Both classes must be added to the csce240_programming_assignment_5 namespace. Classes not added to the csce240_programming_assignment_5 namespace will not be eligible for correctness points.
- You will submit a zip file containing only SongRecording.h, SongRecording.cc, StreamingTrack.h, and StreamingTrack.cc to the assignment in Blackboard

- The only standard library headers that may be included in your code are: iostream and string Submissions that include standard library headers will not be eligible for correctness points.
- Programs must compile and run on a computer of the instructor's choosing in the Linux lab (see your course syllabus for additional details).
- Be sure to review the program expectations section of the course syllabus.

Grade Breakdown

Style *SongRecording.h*: 0.25 points
Style *SongRecording.cc*: 0.25 points
Style *StreamingTrack.h*: 0.25 points
Style *StreamingTrack.cc*: 0.25 points
Documentation: 1 point
Clean compile of *SongRecording.cc*: 0.5 points
Clean compile of *StreamingTrack.cc*: 0.5 points
Passes instructor's modified *testTitle*: 0.5 points
Passes instructor's modified *testTrackLength*: 0.5 points
Passes instructor's modified *testArtist*: 0.5 points
Passes instructor's modified *testSoundRecordingConstructors1*: 0.5 points
Passes instructor's modified *testSoundRecordingCopyConstructor*: 0.5 points
Passes instructor's modified *testSoundRecordingAssignment*: 0.5 points
Passes instructor's modified *testStreams*: 0.5 points
Passes instructor's modified *testAddGenre*: 0.5 points
Passes instructor's modified *testRemoveAndAddGenre*: 0.5 points
Passes instructor's modified *testStreamingTrackConstructors1*: 0.5 points
Passes instructor's modified *testStreamingTrackCopyConstructor*: 0.5 points
Passes instructor's modified *testSoundRecordingAssignment*: 0.5 points
Passes instructor's modified *testStreamingTrackAssignment*: 0.5 points
Passes instructor's modified *testStreamingTrackBaseToDerivedConstructor*: 0.5 points

The penalty for late assignment submissions is 10% per day up to three days after the assignment due date. No assignment submissions will be accepted more that 3 days after the due date.