

# CS 172 Exam 1 (Spring 2017)

You are allowed to look at any of your previous programming solutions, slides, notes, or book for the exam. You may also look at <http://www.cplusplus.com/reference/string/string/>.

We need to work on the honor system. You must put the following statement at the top of any source files you write:

*I affirm that all code given below was written solely by me, <give your name>, and that any help I received adhered to the rules stated for this exam.*

Please save your solution to GitHub repo named CS172-Exam1, and submit the GitHub URL to me via email. If you are unable to submit to GitHub you may email the solution to me. Please note, that submission to GitHub is 10% of your score. So you will lose 10% if you submit via email.

You are writing an application to manage songs in a music playlist for a music player. You will track a set of songs available in a playlist. You will also be able to look for songs in a playlist by name, artist or genre.

Define a class called **Song** and put this class in the file **Song.h**. Remember the **implementation guards**.

- The class has the following **public methods** implemented in a file called **Song.cpp**

```
// default song
Song();
// allow the caller to set up the song.
// valid genres are "ModernPop", "Grunge", "Classical" and "Americana".
Song(string title, string artist, string genre);

string getTitle() const; // return the song title
string getArtist() const; // return the artist who performed the song
string getGenre() const; // return the genre of the song

// return true if the song is of the genre.
bool isOfGenre(string genre);
```

- Think about the additional **private properties** that the **Song** class needs in order to correctly implement these behaviors.

Also define a class called **Playlist**, and put its class definition in a file called **Playlist.h**. Remember the **implementation guards**.

- The playlist class should implement the following **public methods** in a file called **Playlist.cpp**

```
// constructor for the playlist
Playlist(string name);

// get the name of the Playlist
string getName();

// returns true when at least one song of
// this genre is present.
bool hasGenre(string genre);

// Add a new song
void addSong(Song song);

// How many songs in this playlist?
int numberOfSongs();

// Play a song. return an empty string if there is no such song.
string playSong(int songNumber);
```

Once your classes are designed and implemented, test it with the following program:

```
#include <string>
#include <iostream>

#include "Song.h"
#include "Playlist.h"

using namespace std;

int main()
{
    // make an array of songs
    static int numsongs = 9;
    Song songs[] = {
        Song("Human", "Rag'n'Bone Man", "ModernPop"),
        Song("Closer", "The Chainsmokers", "ModernPop"),
        Song("Chained To the Rhythm", "Katy Perry", "ModernPop"),
        Song("Smells Like Teen Spirit", "Nirvana", "Grunge"),
        Song("Black Hole Sun", "Soundgarden", "Grunge"),
        Song("Loser", "Beck", "Grunge"),
        Song("Elvis Presley Blues", "Gillian Welch", "Americana"),
        Song("Pirate's Gospel", "Alela Diane", "Americana"),
        Song("Come On Up To The House", "Sarah Jarosz", "Americana")
    };

    // Make the playlist
    Playlist playlist("iRandomMusic");
    for (int i = 0; i < numsongs; i++)
        playlist.addSong(songs[i]);

    int errors = 0;

    // Does the playlist have the correct number of songs?
    if (numsongs != playlist.numberOfSongs())
    {
        errors++;
    }
}
```

```

        cout << "error: not counting songs correctly\n";
    }

    // Check the genre of one song
    if (playlist.playSong(2) == "")
    {
        errors++;
        cout << "error: genre of the third song is incorrect\n";
    }

    if (playlist.playSong(10) != "")
    {
        errors++;
        cout << "error: index out of range not handled correctly\n";
    }

    // Check for a genre
    if (!playlist.hasGenre("Americana"))
    {
        errors++;
        cout << "error: search for genre is incorrect\n";
    }

    if (errors == 0) cout << "Passed\n";
    else cout << "Errors: " << errors << endl;

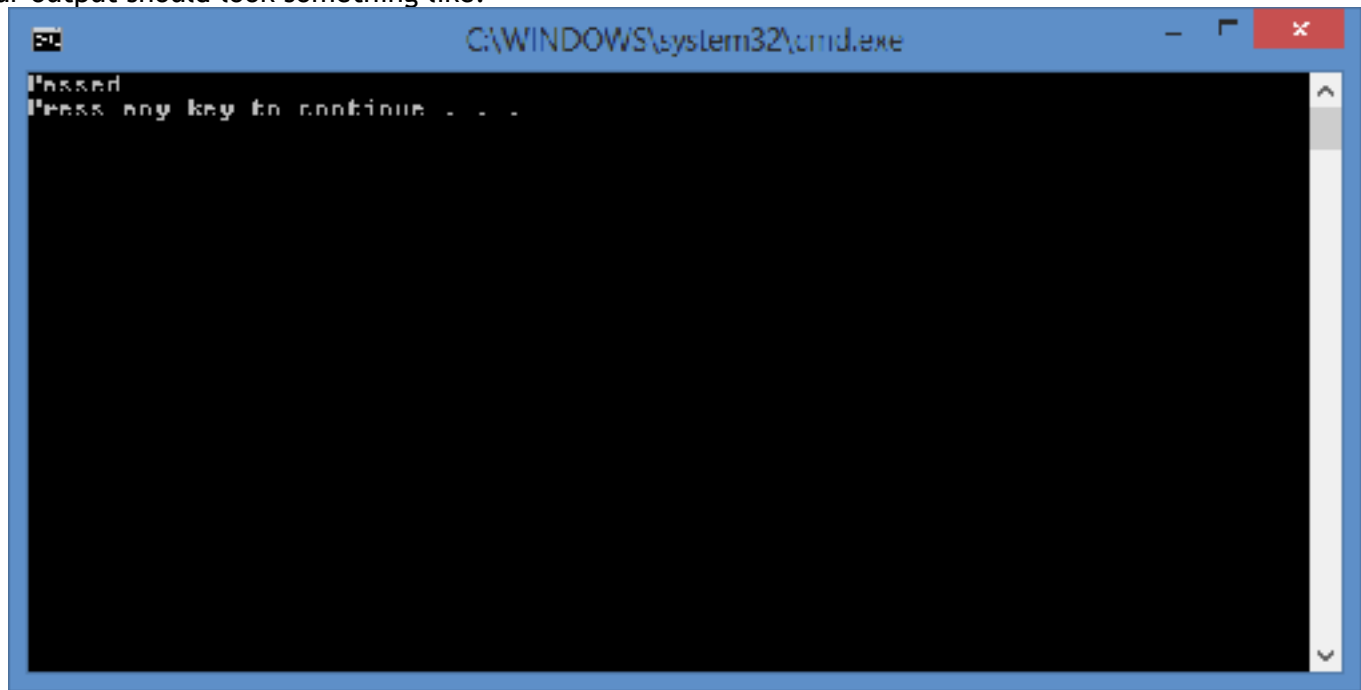
    return errors;
}

```

#### Grade Rubric

Category	Percent
Correct use of member variables and methods.	20%
Correct use of private & public.	10%
Correct Implementation for Song.	20%
Correct Implementation for Playlist.	20%
Comments	20%
Successful GitHub submission	10%

Your output should look something like:



```

C:\WINDOWS\system32\cmd.exe
Passed
Press any key to continue . . .
```

The image shows a Windows command prompt window with a blue title bar. The title bar contains the text "C:\WINDOWS\system32\cmd.exe" and standard window control buttons (minimize, maximize, close). The command prompt area is black with white text. The first line of output is "Passed". The second line is "Press any key to continue . . .". A vertical scrollbar is visible on the right side of the command prompt area.