

Lab 4: Song Class
10 points
DUE: Oct 21 at 11:59PM.

We will be practicing designing a class for a music catalog. Download the file called `songmain.cpp`. **You must use this file as I will be using the same exact file when compiling your code. Please don't change `songmain.cpp` at all as your changes will not be used when I grade your submission.**

Note that `songmain.cpp` makes an array of two `Song` objects. Your task is to design the class `Song`. You need to make two extra files to do this: a `song.cpp` file and a `song.h` file.

The `Song` class has four attributes or fields: `title`, `artist`, `year` and `duration` (in minutes). The `title` and `artist` are C-strings of length 200, `year` is a positive integer and `duration` is a double. These fields should not be public; make them private. Also, implement a full set of setters and getters plus a `print song` function; these should be public. You will also create a default constructor which will set strings to empty and numbers to zero.

HINT: To set a C-string to empty, it is sufficient to just set the first index to the null character:

`title[0] = '\0';`

Makefiles

Look at the slides on `make` in Week 4. Do a makefile for this lab. It doesn't have to be overly complex; the first example in the slides is good enough. I will be grading your lab by compiling and running it with the command `'make'` followed by `'./prog'` so call your executable `'prog'`.

What To Hand In

I just need your `song.cpp`, `song.h`, the makefile, and a recorded `'typescript.txt'` file using the `script` command. The script should be a recording of the compiling (using your makefile) and running of your program. You don't need to hand in `songmain.cpp` as I will be using my own copy of it (it will be the same as the original one on d2l but with different data).

Put these four files in a directory called `'lab4'` and then create a `.tar.gz` file. Upload this to the lab 4 d2l assignment dropbox by the due date.

NOTE: I will not be using the same inputted values as below. So don't 'hard code' to the values below.

```
=====
EXAMPLE OUTPUT (user input is underlined and gray)
=====
```

```
Just after array is created
printing 0 song in array
=====Song=====
title:
artist:
```

```
year: 0
duration in minutes: 0
printing 1 song in array
=====Song=====
title:
artist:
year: 0
duration in minutes: 0
```

```
setting array values and getting values of array
for song 0
title? Hello Hooray
artist? Alice Cooper
year? 1970
duration in minutes? 3.45
for song 1
title? Panic in Detroit
artist? David Bowie
year? 1978
duration in minutes? 4.23
```

```
getting song 0 information with getters
title: Hello Hooray
artist: Alice Cooper
year: 1970
duration: 3.45
```

```
printing song 0 information again
=====Song=====
title: Hello Hooray
artist: Alice Cooper
year: 1970
duration in minutes: 3.45
```

```
getting song 1 information with getters
title: Panic in Detroit
artist: David Bowie
year: 1978
duration: 4.23
```

```
printing song 1 information again
=====Song=====
title: Panic in Detroit
artist: David Bowie
year: 1978
```

duration in minutes: 4.23