

# **Assignment 4 – BCS 230**

**Due: 4/7/2019 @ 11:00pm**

**If there is a real issue out of your control you may submit up until 4/7/2019 @ 11:59pm. “I did not finish doing the assignment” is not a real issue.**

When you submit the assignment in the Blackboard dropbox make sure you zip the whole project and upload the .zip file.

THE FOLLOWING ARE VERY IMPORTANT:

1. The program **MUST** compile correctly. If the program does not compile it cannot run. 50 pts will be taken off if the program does not compile.
2. Make sure you **properly** comment and indent your programs. Download the commenting and indenting documents from Blackboard. They are located in the “Handouts” folder. If you do not properly document your programs points will be deducted.
3. You must zip and submit the **WHOLE** console application project.

## ***Program Overview***

Define and use classes for the song application.

You are responsible for creating any data files used by this program.

Make sure you correctly split up your code into header files (.h) and source files (.cpp). The .h and .cpp files should be properly named. All class definitions should be stored in .h/.cpp pairs. The class definition (member variables and function prototypes) should go in the header file (.h) and the function definitions should go in the source file (.cpp). Include the appropriate preprocessor directives (“guards”) so that the .h file only gets included once. Check the following website for a description of guards: <http://www.learncpp.com/cpp-tutorial/110-a-first-look-at-the-preprocessor/> . Search the webpage for “Header guards”.

If you write any stand-alone functions (a stand-alone function is not inside of a struct or class) they should be stored in a separate .h/.cpp file pair. All function prototypes go in the header file (.h). All function definitions go in the source file (.cpp).

The main function should be in a source file of its own (.cpp).

## ***Part I – Class Definitions***

### **Class – Artist**

#### ***Member Variables (all private)***

<b><u>Variable</u></b>	<b><u>Data Type</u></b>	<b><u>Description</u></b>
Name	String	Contains the first name.
Country of origin	String	Contains the last name.

#### ***Member Function Signatures and Descriptions (all public)***

<b><u>Function Name</u></b>	<b><u>Description</u></b>
Default constructor	Default constructor. Sets the values of each member variable to default values. There are no parameters for this function. DO NOT GET VALUES FROM THE USER INSIDE THIS FUNCTION! DO NOT PRINT ANYTHING IN THIS FUNCTION!
Get/Set methods	Write get/set methods for all the member variables. DO NOT GET VALUES FROM THE USER INSIDE THESE FUNCTIONS! DO NOT PRINT ANYTHING IN THESE FUNCTIONS!

### **Class – Time**

#### ***Member Variables (all private)***

<b><u>Variable</u></b>	<b><u>Data Type</u></b>	<b><u>Description</u></b>
Minutes	Int	Contains the minutes.

Seconds	Int	Contains the seconds.
---------	-----	-----------------------

### ***Member Function Signatures and Descriptions (all public)***

<u>Function Name</u>	<u>Description</u>
Default constructor	Default constructor. Sets the values of each member variable to default values. There are no parameters for this function. DO NOT GET VALUES FROM THE USER INSIDE THIS FUNCTION! DO NOT PRINT ANYTHING IN THIS FUNCTION!
Get/Set methods	Write get/set methods for all the member variables. DO NOT GET VALUES FROM THE USER INSIDE THESE FUNCTIONS! DO NOT PRINT ANYTHING IN THESE FUNCTIONS!

## **Class – Song**

### ***Member Variables (all private)***

<u>Variable</u>	<u>Data Type</u>	<u>Description</u>
Title	String	Contains the song name.
Time	Time	Contains the time information.

### ***Member Function Signatures and Descriptions (all public)***

<u>Function Name</u>	<u>Description</u>
Default constructor	Default constructor. Sets the values of each member variable to default values. There are no parameters for this function. DO NOT GET VALUES FROM THE USER INSIDE THIS FUNCTION! DO NOT PRINT ANYTHING IN THIS FUNCTION!
Get/Set methods	Write get/set methods for all the member variables. DO NOT GET VALUES FROM THE USER INSIDE THESE FUNCTIONS! DO NOT PRINT

	ANYTHING IN THESE FUNCTIONS!
--	------------------------------

## **Class – Album**

### ***Member Variables (all private)***

<b><u>Variable</u></b>	<b><u>Data Type</u></b>	<b><u>Description</u></b>
Title	String	Contains the title of the album.
Artist	Artist	Contains the artist information.
Song Array	Song[6]	An array of songs.

### ***Member Function Signatures and Descriptions (all public)***

<b><u>Function Name</u></b>	<b><u>Description</u></b>
Default constructor	Default constructor. Sets the values of each member variable to default values. There are no parameters for this function. DO NOT GET VALUES FROM THE USER INSIDE THIS FUNCTION! DO NOT PRINT ANYTHING IN THIS FUNCTION!
Get/Set methods for title	Write get/set methods for all the title member variable. DO NOT GET VALUES FROM THE USER INSIDE THESE FUNCTIONS! DO NOT PRINT ANYTHING IN THESE FUNCTIONS!
Time CalcTotalTime()	Returns the amount of time of all the songs in the album.
void Read(string filename)	Reads all data from the given input file. This function must open an ifstream using the filename given in the formal parameter. See below for the layout of the file. DO NOT GET VALUES FROM THE USER INSIDE THIS FUNCTION! READ FROM THE FILE AND POPULATE THE MEMBER VARIABLES.
void Write(string filename)	Writes all data into the given output file. This function must open an ofstream using the filename

	given in the formal parameter. You must write the data according to the file format given below. <u>DO NOT WRITE ANY DESCRIPTIVE TEXT</u> TO THE OUTPUT FILE. THERE SHOULD ONLY BE ONE PIECE OF DATA ON EACH LINE IN THE OUTPUT FILE.
void Display()	Writes all data from all member variables on standard output (the screen). This function should print descriptive text. See the sample execution below.

## ***Part II – User Interface***

### ***Main Function Specifications***

This function should show a menu to the user (description below) and take an action depending on what the user chooses to do. Keep showing the menu and performing an action until the user chooses the Exit menu option.

Required Variables - Declare one instance of type Album inside of main. You may also need other variables.

Here is the menu (sample execution pic at end of document):

**Album Program**

-----

- 1 - Read album info from a file
- 2 - Write album info to a file
- 3 - Show all album info on screen
- 4 - Show album time on screen
- 5 - Exit

***Menu Description***

<u>Choice #</u>	<u>Title</u>	<u>Description</u>
1	Read album info from a file	<p>Fill the Album instance member variables with data from a file. You should ask the user to enter a filename and then read the data from the file that the user specified.</p> <p>Hint: You could use a member function of the Album class to perform most of the work for this menu item.</p>
2	Write album info to a file	<p>Writes all of the Album instance member variable data to a file. You should ask the user to enter a filename and then write all the member variable data to the user specified file. DO NOT WRITE ANY DESCRIPTIVE TEXT TO THE OUTPUT FILE. THERE SHOULD ONLY BE ONE PIECE OF DATA ON EACH LINE.</p> <p>Hint: You could use a member function of the Album class to perform most of the work for this menu item.</p>
3	Show all album info on screen	<p>Write the contents of all member variables of the Album instance to the screen (standard output). This function should contain descriptive text. See the sample execution below.</p> <p>Hint: You could use a member function of the Album class to perform most of the work for this menu item.</p>
4	Show album time on screen	<p>Shows the album time on screen.</p> <p>Hint: You could use a member function of the Album class to perform most of the work for this menu item.</p>

5	Exit	Exit the program.
---	------	-------------------

### ***Album Input File Format***

Title  
 ArtistName  
 CountryOfOrigin  
 SongTitle  
 Minutes  
 Seconds  
 ...  
 SongTitle  
 Minutes  
 Seconds

### ***Album Sample Input Data File (notice there is no descriptive text, just the data)***

Continued Silence  
 Imagine Dragons  
 USA  
 Radioactive  
 3  
 6  
 Demons  
 2  
 57  
 On Top of the World  
 3  
 12  
 Round and Round  
 3  
 17  
 It's Time  
 4  
 0  
 My Fault  
 2  
 56

### ***Sample Execution***

```
C:\WINDOWS\system32\cmd.exe
Album Program
-----
1 - Read album info from a file
2 - Write album info to a file
3 - Show all album info on screen
4 - Show album time on screen
5 - Exit
Enter Choice: 1

Enter input filename: continuedsilence.txt

Album Program
-----
1 - Read album info from a file
2 - Write album info to a file
3 - Show all album info on screen
4 - Show album time on screen
5 - Exit
Enter Choice: 2

Enter output filename: continuedsilencecopy.txt

Album Program
-----
1 - Read album info from a file
2 - Write album info to a file
3 - Show all album info on screen
4 - Show album time on screen
5 - Exit
Enter Choice: 3

Title:      Continued Silence
Artist:     Imagine Dragons
Country:    USA

Radioactive      3:06
Demons           2:57
On Top of the World 3:12
Round and Round  3:17
It's Time       4:00
My Fault        2:56

Album Program
-----
1 - Read album info from a file
2 - Write album info to a file
3 - Show all album info on screen
4 - Show album time on screen
5 - Exit
Enter Choice: 4

Album Time: 19:28

Album Program
-----
```