CS142 **Assignment 4** Spring 2025

Given: Fri, Feb 7
Due: Thu, Feb 13, 3:00 p.m.

1. (24%) Modify the class `Song` of the previous assignment by organizing your code into separate files as explained in Section 2.4 of the notes.

2. (24%) Create a function `print_double_spaced(cs)` that takes a C string as argument and prints the characters of that string separated by a space. For example, the string `hello world` should be printed as

   ```
   h e l l o   w o r l d
   ```

   No space should be printed before the first character of the string or after the last one. The function should print nothing when called on an empty string. Implement the function as efficiently as possible. *Hint*: Don't use `strlen` because that would be inefficient.

3. (28%) Create a program that reads a file containing a list of songs and prints the songs to the screen one at a time. After each song is printed, except for the last song, the program asks the user to press enter for more. After the last song, the program should say that this was the last song and quit. If there were no songs in the file to begin with, the program should say that there are no songs to show and quit.

   The program should begin by asking the user for the name of the input file. Each song consists of a title, artist and year, as described in Assignment 1. In the file, each song is given on three consecutive lines, also as described in Assignment 1.

Create this program using the version of the class `Song` you created for the first exercise of this assignment. Organize your program into separate files as explained in Section 2.4 of the notes.

Hint: Do this exercise gradually. For example, start with a program that just prints all the songs to the screen. Then make it print the songs one at a time, asking the user for more after each song. Then make the program print the correct message if the file is empty. Then figure out how to make it print the right message after the last song.

4. (24%) Create a function `append(v, a, n)` that adds to the end of vector `v` a copy of all the elements of array `a`. For example, if `v` contains 1, 2, 3, and `a` contains 4, 5, 6, then `v` will end up containing 1, 2, 3, 4, 5, 6. The argument `n` is the size of the array. The argument `v` is a vector of integers and `a` is an array of integers. Write a test driver.