CS142 **Assignment 2** Spring 2025

Given: Fri, Jan 24
Due: Thu, Jan 30, 3:00 p.m.

1. (100%) Modify the class `Song` of the previous assignment to make it object-oriented. More precisely, modify the class as follows:

   (a) Replace the function `print(s, out)` by a method `print(out)` that prints the song to output stream `out` in the same format as before. The argument should still have `cout` as a default value.

   (b) Replace the function `read(s, in)` by a method `read(in)` that reads the song from input stream `in`. The data in the stream is assumed to be in the same format as before. The argument should still have `cin` as a default value.

   (c) Replace the function `equals(s1, s2)` by a method `equals(other)` that returns **`true`** if the receiver and the argument have the same title, artist and recording year.

   (d) Add a method `less_than(other)` that returns **`true`** if the title of the first song is less than the title of the second song. In case the titles of the two songs are identical, the method returns **`true`** if the artist of the first song is less than the artist of the second song. In case the titles and artists of the two songs are identical, the method returns **`true`** if the first song was recorded before the second one. In all other cases, the method returns **`false`**. Titles and artists are compared with respect to alphabetical order. (You can use < on strings for that purpose.)

(e) Add a default constructor that initializes the song to have *invalid* as title and artist, and −1 as recording year.

(f) Add a constructor `Song(title)` that initializes the song to the given title (that is, to the title given as argument). The artist and recording year are initialized to *unknown* and −1, respectively. The argument is a string.

(g) Add a constructor `Song(title, artist, year)` that initializes the song to the given title, artist and recording year. The first two arguments are strings, the third one is an integer.

Once again, you'll need to decide how arguments should be passed to the methods.

Declare methods to be constant, as appropriate. Use initializers and delegating constructors, as appropriate.

Write (and submit) a test driver for your class. Make sure you test every method and constructor.