ECE250: Lab Project 0

Due Date: January 17th, 2020, 11:00 pm

1. Project Description

The goal of this project is to write a C++ program that manages a play list of songs. The program allows the user to: (i) add songs to the end of the list, (ii) select a song to be played, and (iii) delete a song from the list. We ask that you use an array to store the play list. Each entry will contain the name of the song, and the corresponding artist.

For this exercise, we ask you to use the object-oriented paradigm to implement your solution. You will create a design where the play list and the entries in the list are represented by classes which: (i) store their properties (data members) and (ii) provide services (function members).

2. Program Design

Write a short description of your design. You will submit this document along with your C++ solution files for marking. This document must include your design decisions. Please refer to the course website for "Programming Guidelines" and the expected content for your design document.

3. Input/Output Requirements

Write a test program named **playlistdriver.cpp** that reads commands from standard input and writes the output to standard output. This program will respond to the commands described in this section.

Command	**Parameters**	**Description**	**Output**
m	*n*	Sets the maximum size of song list	success
i	*s;a* *s:song title* *a:artist*	Adds a song (*song title* and *artist*) at the end of the play list if song is not already in the list and the list is not full	success or can not insert s;a
p	*n* *n: position of a song in the list*	Plays song at position *n*	played n s;a or can not play n
e	*n* *n: position of a song in the list*	Erases song at position *n*	success or can not erase n

- **Test Files**

The course web site contains example input files with the corresponding output. The files are named test01.in, test02.in and so on with their corresponding output files named test01.out, test02.out and so on.

4. How to Submit Your Program

Once you have completed your solution and tested it comprehensively on your own computer or on the lab computers, you have to transfer your files to the *eceUbuntu server* and test there; since we perform the automated testing using this environment. Once you finish testing in the *eceUbuntu server*, you will create a compressed file (tar.gz) that should contain:

- A typed document (maximum two pages) describing your design. Submit this document is PDF format. The name of this file should be:

  xxxxxxxx _design_ p***n***.pdf

  where ***xxxxxxxx*** is your UW user id (e.g., jsmith) and ***n*** is the project number that is 0 (zero) for this submission.

- A test program (playlistdriver.cpp*)* containing your *main()* function
- Required header files and classes (ending in *.h .cpp*)
- A make file (named Makefile), with instructions to compile your solution and create an executable named **playlistdriver**

The name of your compressed file should be ***xxxxxxxx***_p***n***.tar.gz, where ***xxxxxxxx*** is your UW user id (e.g., jsmith) and ***n*** is the project number that is 0 (zero) for this submission.