Fall 2026 COP 3330 Program #10: Make It Your Own (for music/concerts)

Background Information

For program #9, there was some minor ambiguity in the specification (intentionally). The goal here was to have students start making some of their own calls on issues they ran into. A few students seemed to have some difficulty with this. To that end, I'd like to practice giving students more opportunity to make decisions on their own, so this will be a **VERY OPEN-ENDED ASSIGNMENT.**

The assignment will ask you to pursue a few goals with a few minimum requirements. You'll be graded on our perception (TAs and me) of whether or not you achieved those goals. In real life, this is how most things are judged (not on a perfect rubric that you know in advance.)

Assignment Goals

You can design any Java program you want for this assignment with the following goals:

1) Your program must be consistent with this semester's theme of music concerts/bands.

2) Create a fun or useful application using some subset of the tools you've learned this semester.

3) Allow you to practice the skills that will help you do well on the final exam. To get a feel for what might show up on the final exam, feel free to research my archive of past materials for all of my courses.

Basic Minimum Requirements

1) **Don't use AI.** If anything looks too good to be true (ie, not consistent with your exam scores), I'll give it a low grade. (All AI use I catch is because AI does something sophisticated that I know a student couldn't have come up with on their own.)

2) Your program should have multiple classes. One can not begin to appreciate the benefits of object-oriented program design without creating an example with multiple classes.

3) Your program should have some inheritance involved, where one of the classes you create inherits from another class you create. (Same reason as #2.)

4) You have to use at least two of the built in Java API Data Structures that were taught in the course. (ArrayList, ArrayDeque, HashSet, HashMap, TreeSet, TreeMap, PriorityQueue)

5) The total lines of code must exceed 200.

Deliverables
You will submit multiple files:

1. Please submit a .pdf describing what application you've chosen to write, why you chose this, and give the grader appropriate directions on how to use and test your application. (Be clear about which class to run and how we should interact with the program.)

2. Please submit a single .zip file that is a folder containing all of the .java files (and any other files you might need such as image or sound files) that comprise of your application. Please don't create any packages. We'll assume that all source files should reside in the same directory.

Grading Rubric
The TAs and I will have complete discretion on grading broken down into these categories:

Item	Points
Quality of the .pdf file with a focus on the clarity and specificity of the directions to the grader	4
Degree to which both goals were met	4
Basic Minimum Requirements Met (1 pt for each one)	5
Coding Style and Internal Documentation (commenting)	2
Level of complexity and depth relative to the student's expertise	15