

Jefferson Community College

Watertown, NY 13601

Advanced Programming

CIS 216

Spring 2023, section 1

Laboratory 08

1. Write a JAVA program to solve the problem described on the next page. You should not start the actual coding until you have a complete class diagram and system-sequence diagram for the problem. We will create the class diagram and system-sequence diagram together in class.
2. Once you have your class and system-sequence diagrams, you can start entering code for each module, one at a time. Keep in mind that your program should always be a running program. As you finish one module, you will compile the program to test it. If the program compiles and runs, you will start coding the next module. If it doesn't compile and run, then you will need to fix any errors in your coding. This method minimizes the time you will spend finding bugs.
3. When the program is complete, submit a copy. The class diagram, system-sequence diagram, keyboard input, screen output, initial file, ending file, source code (whole zipped project folder), and Laboratory Outcome Summary form should be submitted upon Blackboard.
4. Remember, each program should include the following things (placed in the beginning of the program as comments):
 - Your name
 - Due date
 - Project description
 - Project number

Include comments throughout the program for clarity.

5. Label your class and system-sequence diagrams consistently. Make sure to include comments throughout your program that match the items in the class diagram.
6. Use meaningful identifiers as necessary. Use constants and variables where appropriate. Use good indentation style.
7. Use formatted output statements for ALL numeric values.
8. Use functions and methods to implement the designs exactly. Pass variables where necessary and create local variables where possible.
9. Use objects to implement the program. Properly encapsulate all object attributes.
10. Do not allow any possibility for stale data.
11. ***This is Project 08 and it is due on Tuesday, April 18, 2023***

Problem definition:

Write a program that will implement an audio receiver. The program should take inputs from the keyboard (in place of dials and buttons). Include commands that allow you to turn off the receiver, change band (from AM to FM and vice versa), increment the frequency, decrement the frequency, choose a preset frequency (up to five per band), store current frequency as a preset frequency, increment volume, and decrement volume. When making changes to frequencies, be sure to do so using realistic values. Likewise, when adjusting volumes, ensure that the volumes cannot go outside maximum and minimum ranges. When asking for preset numbers, ensure only valid values are processed.

The program should display the status of the receiver after each input.

Attributes should include band (AM or FM), current frequency AM, current frequency FM, 10 preset channels (5 per band), and volume setting.

The program should run until the user chooses to turn off the receiver. When the program starts, it will read the existing settings from a file. When the receiver is turned off, the program must write the settings back to the same file – you only access the file at these two points.

USE CONSOLE I/O SO THE USER INTERACTIONS CAN BE PRINTED AND SUBMITTED.