Programming Assignment 4

Note: When you turn in an assignment to be graded in this class you are making the claim that you neither gave nor received assistance on the work you turned in (except, of course, assistance from the instructor).

Write a Java class called **Song** to represent a song in a music collection. The data members of the Song class are String objects representing the song **title**, **artist**'s name, and the **album** that includes the song. The instance variables for the class are to have private access modifiers, so you will need to write the appropriate methods to allow a client class to access and modify the data values. In accordance with good programming practice, you are to override the **equals** and **toString** methods inherited from **Object**. In addition, this class is going to be used in a future project collection that is to be sorted, so you'll need to implement the Comparable interface.

A second class is to be written called **SongReader** that contains a main method that reads in a file name via the command line. The file is a listing of the songs that have been formatted using angle-bracketed tags. For example,

```
<song>
    <title>
   The Trapeze Swinger
  </title>
  <artist> Iron & Wine
  </artist>
  <album>
   The Trapeze Swinger - Single
</album>
</song>
```

```
<song>
    <album>
   Melophobia</album>
    <title>
      Cigarette Daydreams
    </title>
    <artist>
      Cage the Elephant
    </artist>
  </song>
```

```
<song>
 <album>
    Young as the Morning,
Old as the Sea </album>
  <title>
    Somebody's Love
  <artist>Passenger
  </artist>
</title>
</song>
```

Each line will a tag (opening or closing), data, or a combination of these elements. Tags are matched pairs of single words surrounded by angle brackets. Opening tags begin with an angle bracket (<) and closing tags begin with an angle bracket and forward slash (</). The tag pairs can appear in any order, but they must be properly nested inside song tags.

Input is read by the **SongReader** class. When reading the input file, data that is not specifically enclosed in song, title, artist, or album tags is to be ignored. The name of the input file will be provided as a command line argument. If the file name is missing or the file cannot be opened, the program should prompt the user to enter a new file name.

Any incorrectly formatted song data should be collected in a report and printed out in an error report to a file called **ErrorLog.txt**. In the example shown above the first two songs are correctly formatted, but the third one is not because the artist tag is inside of the title tag. As another example, in the sample input file, **playlist.data**, the first five song items are valid but the last one is invalid and should be reported as an error by the **SongReader** program.

The **SongReader** class will open and read the input file to create **Song** objects by parsing the file data. You are required to use an implementation of the given **StackInterface<E>** interface in your project for matching opening and closing tags. You do not have to implement the interface, you may use one of the implementations that we have discussed in class and can be found in the Stacks content area of Blackboard. You may not use any external

parsing libraries. After reading the input, the SongReader class will display a list of the valid Song objects. (Note that your code must create a collection of Song objects and iterate over the collection to display the information.)

It is expected that your program will be well documented and you are required to write a private helper method called **printHeading** in the **SongReader** class that outputs the following information to the console in an easy-to-read format: your name, the project number, the course identifier (CMSC 256), and the current semester. You will call this method as the first statement in your main method. All files must contain a comment block at the beginning that includes the file name; all of the same information that was specified for the helper method **printHeading**; and a brief description of the file's purpose.

You will submit the project files (**Song.java** and **SongReader.java**) to the assignment link in Blackboard. Do not compress the files, they should both be submitted as a single submission of two individual files unless Blackboard generates an error message.