**2601 Week 4 Lab**

Create the following class hierarchy:

* **MusicMedia**
  + An abstract class with fields for song title and artist.
  + Constructor that takes parameters to set both fields. Parameters must be validated appropriately before they are assigned to the fields. If either parameter is invalid throw an IllegalArgumentException and do nothing else (let the program terminate).
  + Getter and setter for each field
  + toString() to display the field values
  + Abstract method void play()
* **CompactDisc**
  + Inherits from MusicMedia
  + Has a field to hold the number of tracks on the CD
  + Has a static field called “READING\_METHOD” whose value is “laser”
  + Constructor takes parameters for title, artist. The number of tracks. The number of tracks should be validated appropriately and an IllegalArgumentException will be thrown if the value is bad.
  + Getter and setter for the number of tracks field
  + toString that will display the number of tracks, title, and artist
  + play() method will display a simple message indicating that the cd is being played.
* **Driver class**
  + Will implement main(...)
    - create a CompactDisc object and assign the address to a MusicMedia reference
    - display the objects field values
    - call the play() method.

To this we are going to add two more subclasses and an Interface.

* **Interface FileManager**
  + void save(final String fileName)
  + void delete(final String fileName);
  + more on this later
* **AudioFile**
  + Fields for the file name and the file size in bytes per second (eg. 320 bps)
  + Constructor with parameters for title, artist, filename, file size. File name and file size must be validated appropriately and an IllegalArgumentException will be thrown if the values are bad.
  + Getters, setters toString
  + play() method will display a simple message indicating that the file is being played.
  + Implements FileManager - methods need only output a simple message to the console.
* **VinylRecord**
  + Fields for the number of tracks, the size, and the weight in grams
  + Constructor will take parameters for title, artist, number of tracks, size, and weight. Appropriate validations will be done and IllegalArgumentExeptions will be thrown when values are bad.
  + Valid size will be either 12”, 10” or 7”
  + Valid weight will be either:
    - 140g, 180g, 200g for a 12”
    - 100g for a 10”
    - 40g for a 7” record
  + Getter, setters toString
  + play() method will display a simple message indicating that the record is being played.

Expand the Driver class to create one of each of the MusicMedia types, assigning each address to a MusicMedia reference. Display and the objects’ data to the console and call the various methods you have overridden in the subclasses.

Make sure to use the following in your code:

* initializer block
* this()
* super()
* static initializer block
* static validation method

Submit your project to the dropbox before the deadline.