**2501 Lab 10**

**Purpose:**

In this lab will demonstrate Polymorphism by developing a Music Media library.

**Task:**

1. Create a class hierarchy that features supertype **MusicMedia**, which has fields for musical artist, song title, total number of songs, total runtime, and year. The overloaded constructor will validate all fields and throw an IllegalArgumentException when “bad” data is entered. The class will implement getters for each field, toString(), and the void playSection() method. The playSelection() method will display the following message when called; “Thank you for using our Music Library.”
2. Create these subtypes:
   1. **Record**, which will have additional fields for size of the record in inches, double rpm(revolutions per minute). The constructor will validate these fields to ensure the size is one of 7”, 10”, 12” and throw an IllegalArgumentException for any other value. Similarly, the constructor will validate rpm for 33.3, 45.0 or 78.0. Record will also override both toString() and playSelection(). When called Record.playSelection() will display the output shown below.
   2. **AudioFile**, which will have the additional field for file type. Tthe constructor will validate the to ensure the file type is one of either “mp3”, “m4a” or “wav”. AudioFile will also override both toString() and playSelection(). When called AudioFile.playSelection() will display the output shown below.
   3. **CompactDisc**, which will have additional fields for bonus tracks and digipac, both booleans. CompactDisc will also override both toString() and playSelection(). When called CompactDisc.playSelection() will display the output shown below.
3. Create class MusicLibrary
   1. This class will support an ArrayList<MusicMedia> by implementing the methods void addMedia(MusicMedia media), void displayLibrary(),void playTitle(String title).
      1. void addMedia(MusicMedia media) will add a subtype to the ArrayList, first checking to be sure the parameter is not null
      2. void displayLibrary() will iterate over the ArrayList displaying the entire library using toString()
      3. void playTitle(String title) will validate the parameter appropriately and then search the library for the tile. If found the playSelection() method will be called.
4. Finally, create a driver class that will construct instances of each subtype and add them to the library. Then test the remaining MusicLibrary methods to ensure they work correctly

**Example output:**

Record [size=7, rpm=45.0, toString()=Album [Artist=The Beatles, title=Hey Jude, trackCount=1, totalMinutes=7]]

CompactDisc [bonusTracks=false, digipac=false, toString()=Album [Artist=Neil Young & Crazy Horse, title=Everybody Knows This Is Nowhere, trackCount=4, totalMinutes=40]]

AudioFile [fileType=wav, toString()=Album [Artist=Donnie Iris and the Cruisers, title=Ah Leah!, trackCount=1, totalMinutes=4]]

Thank you for using our Music Library.

You selected the record Hey Jude by The Beatles.

This is a 7 inch record from 1968, playing at 45.0 rpm.

Thank you for using our Music Library.

You selected the CD Everybody Knows This Is Nowhere by Neil Young & Crazy Horse.

This is a Compact Disc from the year 1969.

Thank you for using our Music Library.

You selected the Audio File Ah Leah! by Donnie Iris and the Cruisers.

This file is in wav format, from the year 1980.