

# CSE1322L Assignment 4 - Fall 2024

## Introduction:

A “codec”, short for “Coder/Decoder”, is a program which is used to convert audio and/or video information from one format into another. Originally, codecs were conceived to convert analog signals into digital information for storage in computers, and then back again into analog during media playback. Nowadays, several different libraries are available which implement these codecs, enabling programs to read music or video files.

Codecs may be created with specific goals in mind, such as trying to minimize the amount of space a piece of media occupies on disk, encoding image motion at the expense of color accuracy, or vice-versa. They usually implement some sort of mathematical concept to achieve its goals (e.g.: Discrete Cosine Transform).

In this assignment, we’ll simulate a very simple library which stores information on the media files it receives, including its name, its type, and any codecs it may need. These media files may be either images, music, or videos, and will follow a fictional industry standard, which will be implemented through interfaces.

(It’s worth noting that image files don’t actually have codecs but, instead, have file formats which dictate how the image information must be stored in disk. For the purposes of this assignment, we’ll pretend that images utilize codecs as well.)

**This assignment will make use of a data structure which in Java is called “Array List” and in C# is called “List”. Going forward, both will be simply referred to as a “list”, for simplicity.**

**This assignment requires that a UML diagram is submitted along with your code. UML diagram entities must contain:**

- **The name of the class**
- **All of the classes’ fields and access levels (types are optional)**
- **All of the classes’ methods (constructors are optional)**
- **Arrows indicating the relationship between classes (i.e.: parent and child)**

## Requirements

The features described below must be in your program.

- A total of five classes: Driver, Media, Image, Music, and Video
- A total of 3 interfaces: IMediaStandard, IAudioStandard, and IImageStandard
- Interface IMediaStandard has a single method called “getMediaInfo()”. It takes no parameters and returns a string.

- Interface IAudioStandard is a sub-interface of IMediaStandard and has a single method called “getAudioCodec()”. It takes no parameters and returns a string.
- Interface IImageStandard is a sub-interface of IMediaStandard and has a single method called “getImageCodec()”. It takes no parameters and returns a string.
- Media must be abstract and have the following private fields:
  - A string field named fileName;
  - An integer field named id;
  - A static integer field named nextId;
- Media must have the following constructors:
  - The default constructor sets id to nextId and increments nextId by one.
  - The overloaded constructor takes a String which is used to set fileName. It then sets id to nextId and increments nextId by one.
- Media must have getters for fileName and id.
- Image is a subclass of Media, must implement IImageStandard, and must have the following private field:
  - A string named imageCodec.
- Image must have an overloaded constructor which takes in a String name and a String imageCodec. It must call its parent constructor and pass it “name”, while setting its imageCodec field with the one in the parameters.
- It must implement getImageCodec(), which returns the following string:

Image codec: [IMAGE\_CODEC\_FIELD]

- It must implement getMediaInfo(), which returns the following string:

Image ID: [ID]

Image name: [NAME]

Image codec: [IMAGE\_CODEC\_FIELD]

- Music is a subclass of Media, must implement IAudioStandard, and must have the following private field:
  - A string named audioCodec
- Music must have an overloaded constructor which takes in a String name and a String audioCodec. It must call its parent constructor and pass it “name”, while set its audioCodec field with the one in the parameters.
- It must implement getAudioCodec(), which returns the following string:

Audio codec: [AUDIO\_CODEC]

- It must implement getMediaInfo(), which returns the following string:

Music ID: [ID]

Music name: [NAME]

Audio Codec: [AUDIO\_CODEC\_FIELD]

- Video is a subclass of Media, must implement IImageStandard and IAudioStandard, and must have the following private fields:
  - A string named imageCodec.
  - A string named audioCodec.
- Video must have an overloaded constructor which takes in a String name, a String imageCodec, and a String audioCodec. It must call its parent constructor and pass it “name”, while setting its imageCodec and audioCodec fields with the ones in the parameters.
- It must implement getImageCodec(), which returns the following string:

Image codec: [IMAGE\_CODEC\_FIELD]

- It must implement getAudioCodec(), which returns the following string:

Audio codec: [AUDIO\_CODEC\_FIELD]

- It must implement getMediaInfo(), which returns the following string:

Video ID: [ID]

Video name: [NAME]

Image Codec: [IMAGE\_CODEC\_FIELD]

Audio Codec: [AUDIO\_CODEC\_FIELD]

- The Driver must create a List of Media called allMedia. It must then run a menu in a loop with the following options:
  - Add image: Prompts for a name and an image codec. Creates an instance of an Image with the information prompted and adds it to allMedia.
  - Add music: Prompts for a name and an audio codec. Creates an instance of a Music with the information prompted and adds it to allMedia.
  - Add video: Prompts for a name, an image codec, and an audio codec. Creates an instance of a Video with the information prompted and adds it to allMedia.
  - Show images: Traverses allMedia, calling getMediaInfo() only on instances of Images.
  - Show music: Traverses allMedia, calling getMediaInfo() only on instances of Music.
  - Show videos: Traverses allMedia, calling getMediaInfo() only on instances of Videos.
  - Show images and videos: Traverses allMedia, calling getMediaInfo() only on instances of objects which implement IImageStandard.
  - Show music and videos: Traverses allMedia, calling getMediaInfo() only on instances of objects which implement IAudioStandard.
  - Exit: Terminates the program.

# Considerations

- Remember that you will get partial credit for partial work. Try to deliver as much of the assignment as you can.
- You may add any helper methods you believe are necessary, but you will not get points for them.
- Remember to make full use of inheritance to avoid writing the same lines of code multiple times.
- Proper casting will be necessary to call some of the methods.

## Example: [User input in red]

[Media Manager]

- 1- Add Image
- 2- Add Music
- 3- Add Video
- 4- Show images
- 5- Show music
- 6- Show videos
- 7- Show images and videos
- 8- Show music and videos
- 9- Exit

Enter option: 1

Enter file name: Vacation picture

Enter image codec: JPEG

- 1- Add Image
- 2- Add Music
- 3- Add Video
- 4- Show images
- 5- Show music
- 6- Show videos
- 7- Show images and videos
- 8- Show music and videos
- 9- Exit

Enter option: 2

Enter file name: Country Music

Enter audio codec: FLAC

- 1- Add Image
- 2- Add Music

3- Add Video  
4- Show images  
5- Show music  
6- Show videos  
7- Show images and videos  
8- Show music and videos  
9- Exit  
Enter option: 3

Enter file name: **Wedding**  
Enter image codec: **Motion JPEG**  
Enter audio codec: **WMAL**

1- Add Image  
2- Add Music  
3- Add Video  
4- Show images  
5- Show music  
6- Show videos  
7- Show images and videos  
8- Show music and videos  
9- Exit  
Enter option: 2

Enter file name: **Classical Music**  
Enter audio codec: **TTA**

1- Add Image  
2- Add Music  
3- Add Video  
4- Show images  
5- Show music  
6- Show videos  
7- Show images and videos  
8- Show music and videos  
9- Exit  
Enter option: 1

Enter file name: **Party picture**  
Enter image codec: **PNG**

1- Add Image  
2- Add Music

- 3- Add Video
  - 4- Show images
  - 5- Show music
  - 6- Show videos
  - 7- Show images and videos
  - 8- Show music and videos
  - 9- Exit
- Enter option: 4

Image ID: 1  
Image Name: Vacation picture  
Image codec: JPEG

Image ID: 5  
Image Name: Party picture  
Image codec: PNG

- 1- Add Image
  - 2- Add Music
  - 3- Add Video
  - 4- Show images
  - 5- Show music
  - 6- Show videos
  - 7- Show images and videos
  - 8- Show music and videos
  - 9- Exit
- Enter option: 5

Music ID: 2  
Music Name: Country Music  
Audio codec: FLAC

Music ID: 4  
Music Name: Classical Music  
Audio codec: TTA

- 1- Add Image
- 2- Add Music
- 3- Add Video
- 4- Show images
- 5- Show music
- 6- Show videos

- 7- Show images and videos
- 8- Show music and videos
- 9- Exit

Enter option: 6

Video ID: 3

Video Name: Wedding

Image codec: Motion JPEG

Audio codec: WMAL

- 1- Add Image
- 2- Add Music
- 3- Add Video
- 4- Show images
- 5- Show music
- 6- Show videos
- 7- Show images and videos
- 8- Show music and videos
- 9- Exit

Enter option: 7

Image ID: 1

Image Name: Vacation picture

Image codec: JPEG

Video ID: 3

Video Name: Wedding

Image codec: Motion JPEG

Audio codec: WMAL

Image ID: 5

Image Name: Party picture

Image codec: PNG

- 1- Add Image
- 2- Add Music
- 3- Add Video
- 4- Show images
- 5- Show music
- 6- Show videos
- 7- Show images and videos
- 8- Show music and videos

9- Exit

Enter option: 8

Music ID: 2

Music Name: Country Music

Audio codec: FLAC

Video ID: 3

Video Name: Wedding

Image codec: Motion JPEG

Audio codec: WMAL

Music ID: 4

Music Name: Classical Music

Audio codec: TTA

1- Add Image

2- Add Music

3- Add Video

4- Show images

5- Show music

6- Show videos

7- Show images and videos

8- Show music and videos

9- Exit

Enter option: 9

Shutting down...

## Submitting your answer:

Please follow the posted submission guidelines here:

<https://ccse.kennesaw.edu/fye/submissionguidelines.php>

Ensure you submit before the deadline listed on the lab schedule for CSE1322L here:

<https://ccse.kennesaw.edu/fye/courseschedules.php>