# CSE1322L Assignment 4 - Fall 2025

## 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 requires a UML diagram to be submitted along with your code. UML diagram entities must contain:**

- **The name of the class**
- **All of the classes’ fields and their access levels (types are optional)**
- **All of the classes’ methods (their return types and parameter types are optional. Constructors are also optional)**
- **Arrows indicating the relationship between classes (i.e.: an arrow pointing from Class A to Class B means that Class B is the parent of Class A)**

## Requirements

The features described below must be in your program.

- A total of five classes: the driver, Media, Image, Music, and Video
- A total of 3 interfaces: IMediaStandard, IAudioStandard, and IImageStandard
- Interface IMediaStandard has a single method called “getMediaInfo()”: it has no parameters and returns a string.
- Interface IAudioStandard is a sub-interface of IMediaStandard and has a single method called “getAudioCodec()”: It has no parameters and returns a string.

- Interface IImageStandard is a sub-interface of IMediaStandard and has a single method called “getImageCodec()”: It has no parameters and returns a string.
- Media must be abstract and have the following members:
    - A string field named fileName
    - An integer field named id
    - A static integer field named nextId
    - **Media(String)**: uses the argument to set fileName. It then sets id to nextId and increments nextId by one
    - Getters for fileName and id
- Image is a subclass of Media, must implement IImageStandard, and must have the following members:
    - A string field named imageCodec.
    - **Image(String, String)**: The first argument must be passed to its superclass constructor while the second argument must be used to set “imageCodec”
    - **String getImageCodec():** This override returns the following string:

      ```
      Image codec: {imageCodec}
      ```

    - **String getMediaInfo()**: This override returns the following string:

      ```
      Image ID: {id}
      Image name: {name}
      Image codec: {imageCodec}
      ```

- Music is a subclass of Media, must implement IAudioStandard, and must have the following members:
    - A string field named audioCodec
    - **Music(String, String)**: The first argument must be passed to its superclass constructor. The second argument is used to set “audioCodec”
    - **String getAudioCodec()**: This override returns the following string:

      ```
      Audio codec: {audioCodec}
      ```

    - **String getMediaInfo()**: This override returns the following string:

      ```
      Music ID: {id}
      Music name: {name}
      Audio Codec: {audioCodec}
      ```

- Video is a subclass of Media, must implement IImageStandard and IAudioStandard, and must have the following members:
    - A string field named imageCodec
    - A string field named audioCodec

- **Video(String, String, String)**: The first argument must be passed to its superclass constructor. The second and third argument must be used to set imageCodec and audioCodec, respectively
- **String getImageCodec()**: This override returns the following string:

```
Image codec: {imageCodec}
```

- **String getAudioCodec()**: This override returns the following string:

```
Audio codec: {audioCodec}
```

- **String getMediaInfo()**: This override returns the following string:

```
Video ID: {id}
Video name: {name}
Image Codec: {imageCodec}
Audio Codec: {audiCodec}
```

- The Driver must create an arraylist 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.

## Deliverables

- Media.java
- Image.java
- Music.java
- Video.java
- IMediaStandard.java
- IAudioStandard.java
- IImageStandard.java
- Assignment4.java (driver)
- UML.pdf (UML diagram)

## Considerations

- Remember that you will get partial credit for partial work. Try to deliver as much of the assignment as you can.
- Contrary to what the deliverables above say, you can submit all of your classes and interfaces in a single file.
- To make an interface a sub-interface of another interface, simply use the “extends” keyword much like as you would trying to make a class a subclass of another class
    - Note that a class that implements a sub-interface must also implement the methods required by the super-interface
- 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.

## Sample Output (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...
```