

# Lab Activity and Assignment #2

## 1 Introduction

You just got an internship at Netfliz, a streaming video company. Great! Your first assignment is to create an application that helps the users to get facts about their streaming videos. The company works with TV Series and also Movies. Your app shall display simple dialog boxes and help the user to make the choice of what to see. An example of such navigation is shown below:

**Path #1: Customer wants to see facts about a movie:**

Input

Netflix App

- 1) Movies
- 2) TV Series
- 3) Exit

Please select one option:

OK Cancel

>>

Input

List of Movies:

- 1) Beauty and the Beast
- 2) The Matrix
- 3) Exit

Please select one option:

OK Cancel

>>

Message

**i** Movie Title: Beauty and the Beast  
Viewer Rating: PG  
Description: Beauty and the Beast is a 2017 American musical romantic fantasy film directed by Bill Condon from a  
Release year 2017  
Runtime: 129 minutes  
Production Company: Walt Disney  
Casting: Emma Watson, Dan Stevens, Luke Evans, Kevin Kline, Josh Gad, Ewan McGregor, Stanley Tucci, Audr

**Path #2: Customer wants to see facts about a TV Series:**

Input

Netflix App

- 1) Movies
- 2) TV Series
- 3) Exit

Please select one option:

OK Cancel

>>

Input

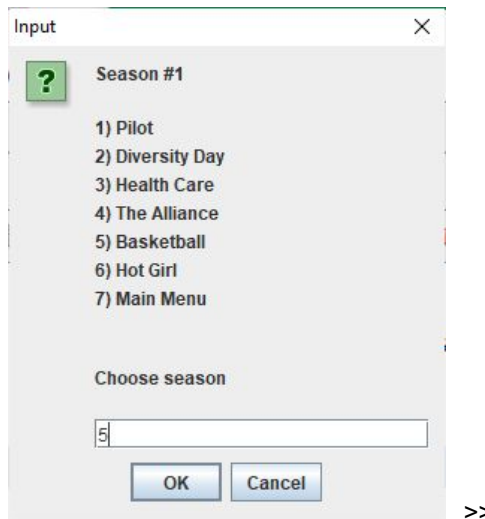
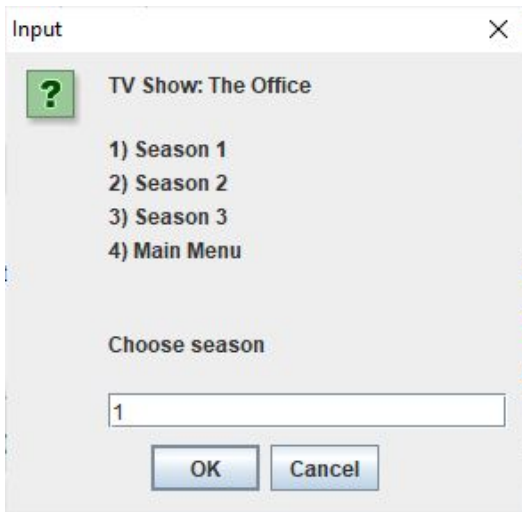
List of Shows:

- 1) Breaking Bad
- 2) The Office
- 3) Exit

Please select one option:

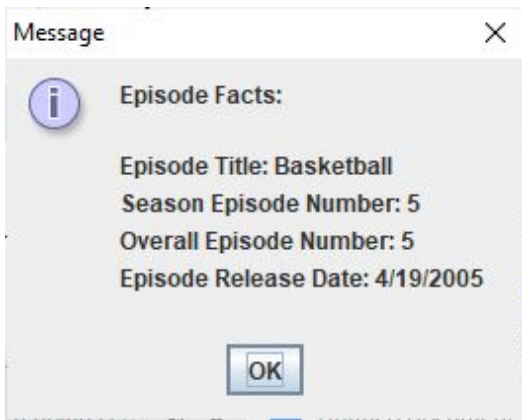
OK Cancel

>>



>>

>>



Your app shall read the facts about a Movie or a TV Show from text files (in some other course you will learn how to retrieve this information from a database). They are provided at the end of this document.

As part of your lab, you should be creating all the classes up to Section 3 (inclusive). As part of your lab you should be creating the main Netflix App and making sure that your code does as shown in the figures above. The Assignment is due on March 8th.

By doing this activity, you should be practicing the concept and application of the following Java OOP concepts

Class Fields	Class Methods	Getter methods	Setter methods	encapsulation	Lists
String class	Split methods	Reading text Files	Scanner class	toString method	Override superclass methods
Scanner Class	JOptionPane Class	Super-class	sub-class	Inheritance	polymorphism
Object Aggregation	Class Constructors	Private methods	Public methods	FOR loops	WHILE Loops
Extending Super class methods	StringBuilder	Variables	IF statements	User Input	And much more...

## 2 Overall App Structure

Any given movie has the following available information:

- Title
- Viewer Rating
- Description
- Release year
- Runtime
- Production Company
- Cast

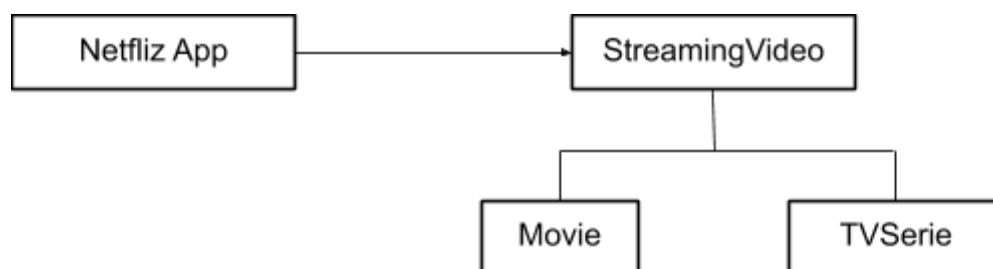
And any given TV Series has the following available information:

- Title
- Viewer Rating
- Description
- Cast
- Number of Seasons
- For each Season:
  - Overall Series Episode
  - Season Episode
  - Episode Title
  - Original Released Year

Since there are many common features between Movies and TV Series, you will create a super class (StreamingVideo) that will contain the common features and then create two subclasses, one for movies and another for TV Series that will extend the basic functionality of the StreamingVideo).

The TV Series is a little bit more complex than the Movie class because it will contain a list of seasons and each season has a list of episodes. Seasons and Episodes objects will be created based on Java classes Season and Episode respectively.

By using polymorphism, your Netflix App will manage multiple (and of different types of) Streaming videos, such as Breaking Bad (TV Series), The Office (TV Series), the Beauty and the Beast (movie), and The Matrix (movie).



## 3 App Classes

### 3.1 The StreamingVideo Class

The following fields were identified as common fields for both Movies and TV Series, so we will place them in the superclass StreamingVideo:

- private String title;
- private String rating;
- private String description;
- private List<String> casting;

The fields are private, therefore create getters and setters to be able to access/modify them during run time. No need for constructors in this class. We will create Movies and TV Series objects that are subclasses of this class.

### 3.2 The Movie Class

The Movie class shall extend the functionality of the StreamingVideo class. It also shall have movie specific class fields as shown below (these features are not common to the TV Series).

- private String runTime;
- private String productionCompany;
- private String releasedYear;

Create getter and setter methods for the above class fields.

There must be one single Class Constructor that takes a single input argument: the location of a specific movie text file, such as "beauty\_and\_the\_beast.txt". This constructor shall:

- Open the file using a Scanner class
- Get the movie title, rating, description, release year, runtime, production company, and casting. Place all this info either in the inherited fields (from StreamingVideo class defined/inherited fields) or in the Movie specific class fields.

Create a toString() method that creates a String to be used as part of the message that we will put in the JOptionPane dialog box later on. The information in the String should be the following:

- Movie Title:
- Viewer Rating
- Description
- Release year
- Runtime
- Production Company
- Casting

You must use StringBuilder to create this String with all this information.

## 3.3 The TVShow Class

The TVShow class shall extend the functionality of the StreamingVideo class. It also shall have TV series specific class fields as shown below (these features are not common to Movies).

- private int numberOfSeasons;
- private List<Season> listOfSeasons;

Note: you might need to stop implementing the TVShow class for a few minutes and implement the Season and Episode classes first, since TVShow class depends on them, and then come back and finish this class.

Create getter and setter methods for the above class fields.

There must be one single Class Constructor that takes a single input argument: the location of a specific TV series text file, such as "breathing\_bad.txt". This constructor shall:

- Open the file using a Scanner class
- Get the TV Series title, rating, description, casting, seasons information. Place all this info either in the inherited fields (from StreamingVideo class defined/inherited fields) or in the specific class fields.

Note: Reading the Seasons is a little more complex. Please look at the Breaking Bad text file to verify how the seasons and episodes are given in the text file, so that you can implement the java code to retrieve correctly:

- Reset the listOfSeasons class field by "listOfSeasons = new ArrayList();"
- Loop over each season:
  - Create a Season object
  - Set season number
  - Create a list of Episodes (List<Episode>)
  - Loop over all episodes of the current season, creating Episode objects, and adding them to the list of Episodes
- After reading all the episodes of a given season, add the list of episodes into the season object
- Add this current season object to the list of Seasons from this class object (TV show object)

Create a toString() method that creates a String to be used as part of the message that we will put in the JOptionPane dialog box later on. The information in the String should be the following:

- TV Show
- List of series with a number in front of it, so the user can select what series to select

Use a StringBuilder object to create the above string.

## 3.4 The Season Class

The Season class does not extend any class. Its class fields are shown below:

- private int seasonNumber;
- private List<Episode> listOfEpisodes;

Create getter and setter methods for the above class fields.

You can create one constructor that takes the necessary input to create a Season object or you can avoid creating the class constructor and using the setters and getters to fill the class fields. I leave it up to you.

Create a toString() method that creates a String to be used as part of the message that we will put in the JOptionPane dialog box later on. The information in the String should be the following:

- Season #
- List of episodes for this specific season with a number in front of it, so the user can select what series to select

Use a `StringBuilder` object to create the above string.

## 4 The Netflix App

All the Java classes that you have created in Section 3 above are just blueprints, don't forget about that. You need an application to create actual objects (instances of) those blueprints (classes). This is the main purpose of the Netflix app: it reads the text files, creates instances of TV shows and Movies (with TV shows with instances of Seasons and Episodes).

The Netflix app will also be responsible for creating all the `JOptionPane` dialog boxes described in Section 1 of this document.

The following tasks are to be implemented in your Netflix app:

```
public class NetflixApp {
    public static void main(String[] args) {
        new NetflixApp();
    }
    public NetflixApp() {
        // reading all movies and tv series from text file (simulating a database)
        List<StreamingVideo> listOfStreamingVideos = readStreamVideoDatabase();
        showMainMenu(listOfStreamingVideos);
    }

    private List<StreamingVideo> readStreamVideoDatabase() {
        List<StreamingVideo> streamVideoList = new ArrayList();
        File databaseFile = new File("StreamVideoDatabase.txt");
        try {
            Scanner fileScanner = new Scanner(databaseFile);
            while (fileScanner.hasNext()) {
                String line = fileScanner.nextLine();
                String[] items = line.trim().split("=");
                if (items[0].trim().equalsIgnoreCase("tv_show")) {
                    streamVideoList.add(new TVShow(items[1].trim()));
                } else {
                    streamVideoList.add(new Movie(items[1].trim()));
                }
            }
        } catch (FileNotFoundException ex) {
            Logger.getLogger(NetflixApp.class.getName()).log(Level.SEVERE, null, ex);
        }
        return streamVideoList;
    }

    private void showMainMenu(List<StreamingVideo> listOfStreamingVideos) {
        // implement your code here (use the toString method when possible)
    }

    private void showMoviesMenu(List<StreamingVideo> listOfStreamingVideos) {
        // implement your code here (use the toString method when possible)
    }

    private void showSpecificMovieInfo(Movie movie) {
        // implement your code here (use the toString method when possible)
    }

    private void showTVShowMenu(List<StreamingVideo> listOfStreamingVideos) {
        // implement your code here (use the toString method when possible)
    }

    private void showSpecificTVShowMenu(TVShow tvShow) {
        // implement your code here (use the toString method when possible)
    }

    private void showSpecificTVShowSeason(Season tvShowSeason) {
        // implement your code here (use the toString method when possible)
    }
}
```

```
private void showSpecificTVShowSeasonEpisode(Episode episode) {  
    JOptionPane.showMessageDialog(null, episode);  
}
```

The above green-highlighted code is an example of using the `toString()` method from the Class `Episode`. When using the object's name without anything else, Java Virtual Machine calls the `toString()` instead of printing the memory address as we discussed in class



# Appendix A:

## StreamVideoDatabase.txt

Create a text file in your Netbeans project under the name StreamVideoDatabase.txt and copy and paste the following contents. Just remember to update the file path to what you have in your computer.

```
tv_show = <your_file_path_here>\\BreakingBad.txt
tv_show = <your_file_path_here>\\TheOffice.txt
movie = <your_file_path_here>\\BeautyAndTheBeast.txt
movie = <your_file_path_here>\\TheMatrix.txt
```

# Appendix B:

## BeautyAndTheBeast.txt File

Create a text file in your Netbeans project under the name BeautyAndTheBeast.txt and copy and paste the following contents. Just remember to update the file path to what you have in your computer.

Title: Beauty and the Beast

Viewer Rating: PG

Description: Beauty and the Beast is a 2017 American musical romantic fantasy film directed by Bill Condon from a screenplay written by Stephen Chbosky and Evan Spiliotopoulos. Co-produced by Walt Disney Pictures and Mandeville Films,[1][7] it was filmed entirely in Britain with predominantly British principal actors. The film is a live-action adaptation of Disney's 1991 animated film of the same name, itself an adaptation of Jeanne-Marie Leprince de Beaumont's 18th-century fairy tale.[8] It features an ensemble cast including Emma Watson and Dan Stevens as the eponymous characters, with Luke Evans, Kevin Kline, Josh Gad, Ewan McGregor, Stanley Tucci, Audra McDonald, Gugu Mbatha-Raw, Ian McKellen, and Emma Thompson in supporting roles.

Release year: 2017

Runtime: 129 minutes

Production Company: Walt Disney

Cast: Emma Watson; Dan Stevens; Luke Evans; Kevin Kline; Josh Gad; Ewan McGregor; Stanley Tucci; Audra McDonald; Gugu Mbatha-Raw; Ian McKellen

# Appendix C

## TheMatrix.txt File

Create a text file in your Netbeans project under the name TheMatrix.txt and copy and paste the following contents. Just remember to update the file path to what you have in your computer.

Title: The Matrix

Viewer Rating: PG-13

Description: The Matrix is a 1999 science fiction action film[3][4] written and directed by the Wachowskis.[a] It stars Keanu Reeves, Laurence Fishburne, Carrie-Anne Moss, Hugo Weaving, and Joe Pantoliano and is the first installment in the Matrix franchise. It depicts a dystopian future in which humanity is unknowingly trapped inside a simulated reality, the Matrix, created by intelligent machines to distract humans while using their bodies as an energy source.[5] When computer programmer Thomas Anderson, under the hacker alias "Neo", uncovers the truth, he "is drawn into a rebellion against the machines"[5] along with other people who have been freed from the Matrix.

Release year: 1999

Runtime: 136 minutes

Production Company: Warner Bros

Cast: Keanu Reeves; Laurence Fishburne; Carrie-Anne Moss; Hugo Weaving; Joe Pantoliano

# Appendix D

## TheOffice.txt File

Create a text file in your Netbeans project under the name TheOffice.txt and copy and paste the following contents. Just remember to update the file path to what you have in your computer.

Title: The Office

Viewer Rating: PG-13

Description: The Office is an American television sitcom broadcast on NBC. Created as an adaptation by Greg Daniels of the British series of the same name, it is a mockumentary that follows the day-to-day lives of the employees of the Scranton, Pennsylvania branch of Dunder Mifflin, a fictional paper supply company.[1] The series ran on NBC in the United States from March 24, 2005 to May 16, 2013. Additionally, nine spin-off series of webisodes of The Office have been aired on NBC.com.

Cast: Steve Carell; Rainn Wilson; John Krasinski; Jenna Fischer; B. J. Novak

Number of Seasons: 3

Season: 1

1;1;Pilot;3/24/2005

2;2;Diversity Day;3/29/2005

3;3;Health Care;4/5/2005

4;4;The Alliance;4/12/2005

5;5;Basketball;4/19/2005

6;6;Hot Girl;4/26/2005

Season: 2

7;1;The Dundies;9/20/2005

8;2;Sexual Harassment;9/27/2005

9;3;Office Olympics;10/4/2005

10;4;The Fire;10/11/2005

11;5;Halloween;10/18/2005

12;6;The Fight;11/1/2005

13;7;The Client;11/8/2005

14;8;Performance Review;11/15/2005

15;9;Email Surveillance;11/22/2005

16;10;Christmas Party;12/6/2005

17;11;Booze Cruise;1/5/2006

18;12;The Injury;1/12/2006

19;13;The Secret;1/19/2006

20;14;The Carpet;1/26/2006

21;15;Boys and Girls;2/2/2006

22;16;Valentine's Day;2/9/2006

23;17;Dwight's Speech;3/2/2006

24;18;Take Your Daughter to Work Day;3/16/2006

25;19;Michael's Birthday;3/30/2006

26;20;Drug Testing;4/27/2006

27;21;Conflict Resolution;5/4/2006

28;22;Casino Night;5/11/2006

Season: 3

29;1;Gay Witch Hunt;9/21/2006

30;2;The Convention;9/28/2006

31;3;The Coup;10/5/2006

32;4;Grief Counseling;10/12/2006

33;5;Initiation;10/19/2006

34;6;Diwali;11/2/2006

35;7;Branch Closing;11/9/2006

36;8;The Merger;11/16/2006  
37;9;The Convict;11/30/2006  
38;10;A Benihana Christmas;12/14/2006  
40;12;Back from Vacation;1/4/2007  
41;13;Traveling Salesmen;1/11/2007  
42;14;The Return;1/18/2007  
43;15;Ben Franklin;2/1/2007  
44;16;Phyllis' Wedding;2/8/2007  
45;17;Business School;2/15/2007  
46;18;Cocktails;2/22/2007  
47;19;The Negotiation;4/5/2007  
48;20;Safety Training;4/12/2007  
49;21;Product Recall;4/26/2007  
50;22;Women's Appreciation;5/3/2007  
51;23;Beach Games;5/10/2007  
52;24;The Job;5/17/2007

# Appendix E

## BreakingBad.txt File

Create a text file in your Netbeans project under the name BreakingBad.txt and copy and paste the following contents. Just remember to update the file path to what you have in your computer.

Title: Breaking Bad

Viewer Rating: TV-MA

Description: Breaking Bad is an American television drama series created by Vince Gilligan, which premiered in 2008 on the cable network AMC. It revolves around Walter White (Bryan Cranston), a 50-year-old high school chemistry teacher in Albuquerque, New Mexico. After White is diagnosed with lung cancer, he uses his chemistry knowledge to cook crystal meth with his former student Jesse Pinkman (Aaron Paul), in order to secure his family's (played by Anna Gunn and RJ Mitte) financial future before he dies from the disease.

Cast: Bryan Cranston; Anna Gunn; Aaron Paul; Dean Norris; Betsy Brandt; RJ Mitte

Number of Seasons: 3

Season: 1

1;1;Pilot;1/20/2008

2;2;Cat's in the Bag...;1/21/2008

3;3;...And the Bag's in the River;1/22/2008

4;4;Cancer Man;1/23/2008

5;5;Gray Matter;1/24/2008

6;6;Crazy Handful of Nothin';1/25/2008

7;7;A No-Rough-Stuff-Type Deal;1/26/2008

Season: 2

8;1;Seven Thirty-Seven;3/8/2009

9;2;Grilled;3/15/2009

10;3;Bit by a Dead Bee;3/22/2009

11;4;Down;3/29/2009

12;5;Breakage;4/5/2009

13;6;Peekaboo;4/12/2009

14;7;Negro y Azul;4/19/2009

15;8;Better Call Saul;4/26/2009

16;9;4 Days Out;5/3/2009

17;10;Over;5/10/2009

18;11;Mandala;5/17/2009

19;12;Phoenix;5/24/2009

20;13;ABQ;5/31/2009

Season: 3

21;1;No Más;3/21/2010

22;2;Caballo sin Nombre;3/28/2010

23;3;I.F.T.;4/4/2010

24;4;Green Light;4/11/2010

25;5;Mas;4/18/2010

26;6;Sunset;4/25/2010

27;7;One Minute;5/2/2010

28;8;I See You;5/9/2010

29;9;Kafkaesque;5/16/2010

30;10;Fly;5/23/2010

31;11;Abiquiu;5/30/2010

32;12;Half Measures;6/6/2010

33;13;Full Measure;6/13/2010