

ICS 45J: Programming in Java – Winter 2017

Lab 1

Deadline: 1/19/2017

For this homework, I want students to get familiar with the structure of Java programs and the Eclipse environment. You will work on writing classes, using getters / setter methods, constructors, and accessing modifying the state of objects in your program. If you have not completed Lab 0 by now, please do so. You will need to get Java and Eclipse up and running.

I want you to write a simple program that can represent the properties of a typical blu-ray movie. The structure of your program will consist of the following components:

- A Movie class consisting of the following private attributes:
 - o int - duration of the movie in minutes
 - o int - year the movie was released
 - o double - price of the movie in USD
 - o String - title of the movie
 - o Director – an object representing the Director of the movie(for this assignment, you may assume the movie has only one director).
- An Director class consisting of the following private attributes:
 - o String - first name of Director
 - o String – last name of Director
 - o int – birth year of Director
 - o int – number of films the director has directed
- A Lab1 class consisting of:
 - o The main method that starts your Java program.

You will write the appropriate accessor / mutator methods for all attributes in the classes as well as default and copy constructors. For the Movie class, you will override the method `.toString()` which prints out all of the information of the Movie including the details of the Director. Note: when writing the setter method for the Director attribute, you will need to pass in the appropriate fields so the Director object can be constructed. For example:

```
public void setDirector(String firstName, String lastName, int birthYear,
                        int numOfDirectedMovies) { ... }
```

Your main method should contain code that illustrates the following functionality:

- Construct a Movie object using the default constructor.
- Set the appropriate fields using the Movie's setter methods.
- Print out the Movie's information (including the Directors's information) by calling the Movie's `.toString()` method.

Test that your copy constructor works by doing the following:

- Create another movie that is a copy of your original Movie object using the copy constructor.
- Print out the fields of the copied object to the console and confirm that all of the fields are the same as the original object.

Change the state of the object by:

- Increment the number of releases for the Director of the Movie.
- Print out the copied object with the updated number of releases.

In your main method, you can “hard-code” the values in the parameters when initially constructing your movie object. For example, to construct the object you can do the following:

```
// Create a new movie with default values for all
attributes Movie movie = new Movie();

// Set the appropriate fields for the book's objects
movie.setTitle("Harry Potter and the Goblet of
Fire"); movie.setPrice(19.99);
movie.setYearReleased(2000);
movie.setDurationInMinutes(734);
movie.setDirector("J.K.", "Rowling", 1965, 7);

// Prints out the state of the movie.
System.out.println(movie.toString());
```

An example output of the following code snippet is:

```
Title: Harry Potter and the Goblet of Fire
Published in: 2000
Duration of Movie: 330 minutes
Price: $19.99
Directed by J.K. Rowling, who was born in 1965 and has 7
releases.
```

Note: This is not the only way to construct your Java program (there are MANY ways to design your code to solve a problem). As a side-exercise, think of other ways you can structure your program to accomplish the same thing. For example, is it possible to allow the main method to construct an Director object and pass this object when setting the Director attribute in the Movie class? What are the advantages / disadvantages of this? Can you accomplish copying an object without copy constructors? If so, why would you want copy constructors?