**CMSC203 Lab1 – Driver and Data Element**

**Driver to test a class**

**Lab Objectives**

* Write a Java program from pseudo-code
* Write a driver program to test a separate class
* Create an object from a class’s constructor
* Write a loop to repeat a task

**Introduction**

In this lab, you are introduced to multiple classes (a driver class and a data element class). You will write the driver class in order to test the various methods provided in the data element class.

You are given a file called Movie.java, which has the data fields for a movie, along with “setters” and “getters”, and a “toString” method. You will create a driver class from the pseudocode in Task #1 below to test the Movie class.

Follow the directions exactly, and in order.

**Task #0 Review the Movie Class**

1. Copy the file Movie.java (see code listing 3.1) from Blackboard.
2. Create a project in Eclipse and import Movie.java.
3. Open Movie.java and observe the three data fields (attributes). Notice that the fields are declared as private, so they are not accessible outside the class. They are accessed via public setter and getter methods and in general any public method defined within the class Movie. (This is called “data hiding”, a part of “encapsulation”, and a good programming practice). There is also a method in Movie called toString(), which prints out the information for a movie.
4. **NOTE**: You should not modify Movie.java.

**Task #1 Writing a Driver Class**

1. Create a new class called MovieDriver. In Eclipse, right-click on the project you created and select New->Class. In the wizard, name the class MovieDriver and check the box to create a main method.
2. Open the file MovieDriver.java in Eclipse. Write Java code to implement the following pseudocode:

*Create a new object of type Scanner that reads from the keyboard*

*Create a new movie object*

*Prompt the user to enter the title of a movie*

*Read in the line that the user types*

*Set the title in the movie object*

*Prompt the user to enter the movie’s rating*

*Read in the line that the user types*

*Set the rating in the movie object*

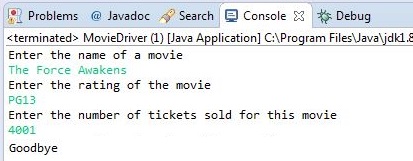
*Prompt the user to enter the number of tickets sold at a (unnamed) theater*

*Read in the integer that the user types*

*Set the number of tickets sold in the movie object*

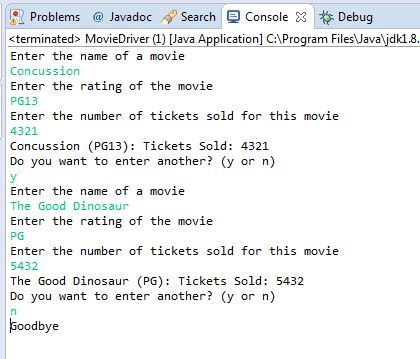
*Print out the information using the movie’s toString method*

1. Run your driver class to be sure it prints out the information you type in.
2. Your Eclipse console should look something like this:



**Task #2 Writing a Loop**

1. Add to your driver class a loop that reads input for multiple movies. Your code does not need to save each movie as you enter the next one.
2. Ask the user if they want to continue, and continue the loop if the user types the correct response.
3. Run the new driver to ensure that it reads and prints multiple movies.
4. Your Eclipse console should look something like this:



1. HINT: if your program ends before reading your keyboard input, you may have not read the previous line feed, so when you read a line expecting it to be the next line of input, it is instead just the previous “\n”. To solve this problem, you may be able to put in an additional keyboard.nextLine(); just to read and discard the line feed.;where keyboard is an object of the scanner class.

**Task #3 – Upload your java files to GitHub**

Upload your java files to GitHub in a directory named CMSC203\_Lab1.

**Turn in the following:**

**Task1:**

* MovieDriver.java from Task #1 and Movie.java (**NOT the .class files**; submit the .java files). Submit both files, even though you have not changed Movie.java.
* Screenshot of the running output

**Task2:**

* MovieDriver.java from Task #2 and Movie.java (**NOT the .class files**; submit the .java files). Submit both files, even though you have not changed Movie.java.
* Screenshot of the running output

**Task3:**

* Screenshot of the uploaded your java files to GitHub in a directory named CMSC203\_Lab1

**Code Listing 3.1 – Movie.java**

public class Movie {

private String title;

private String rating;

private int soldTickets;

public Movie ()

{

title = "";

rating = "";

soldTickets = 0;

}

public Movie (Movie m)

{

title = m.title;

rating = m.rating;

soldTickets = m.soldTickets;

}

public Movie(String title, String rating, int soldTickets) {

this.title = title;

this.rating = rating;

this.soldTickets = soldTickets;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getRating() {

return rating;

}

public void setRating(String rating) {

this.rating = rating;

}

public int getSoldTickets() {

return soldTickets;

}

public void setSoldTickets(int soldTickets) {

this.soldTickets = soldTickets;

}

public String toString() {

return (this.title+" ("+this.rating+"): Tickets Sold: "+this.soldTickets);

}

}