Lab 8 Name:_________________________ Checked:______

Preparation:

Review the textbook example of a DVD database. Use this program as a model to create a database to store data about passengers of the Titanic. File links:

- www.csc.villanova.edu/~map/1051/Chap08/DVD.java
- www.csc.villanova.edu/~map/1051/Chap08/DVDCollection.java
- www.csc.villanova.edu/~map/1051/Chap08/Movies.java
- Dataset: http://www.csc.villanova.edu/~map/2014/f14/examples/titanic.txt

Classes:

You should create the following classes:

1) `Passenger` class:

Represents a passenger of the Titanic, with attributes:

- **`status`** (an integer: 1, 2, 3, or 4, representing 1st, 2nd, 3rd class or crew
- **`child`** (a boolean: true = child, false = adult)
- **`sex`** (a String: “male” or “female”)
- **`survivor`** (a boolean: true/false indicating whether this passenger survived)

The constructor and toString() methods should work in a way similar to the DVD class.

2) `PassengerData` class: Represents a collection of **`Titanic`** passenger records, similar to the **`DVDCollection`** class. Attributes should include:

- **`collection`** (an array of **`Passenger`**)
- **`count`** (an **`int`** representing the size of the database)
- **`numSurvivors`** (an **`int`** representing the number of survivors)

The constructor and other methods should parallel the ones in **`DVDCollection`**.

3) `TitanicTester` class: This is a test client, similar to **`Movies`**.

```
public class TitanicTester
{
   public static void main(String[] args)
   {
      TitanicData titanic = new TitanicData();

      titanic.addPassenger (4, false, “male”, false);
      titanic.addPassenger (3, false, “female”, false);
      titanic.addPassenger (1, true, “male”, true);
      titanic.addPassenger (2, false, “male”, false);
      titanic.addPassenger (3, true, “female”, true);

      System.out.println(titanic);

      titanic.addPassenger (3, true, “male”, false);
      titanic.addPassenger (1, true, “female”, false);
      titanic.addPassenger (3, false, “female”, false);
      System.out.println(titanic);
   }
}
```

Before proceeding, be sure that **`TitanicTester`** runs correctly (modify your **`Passenger`** and **`PassengerData`** classes, if necessary).

4) `Titanic` class: Create another client, similar to **`TitanicTester`** to access and process a data file with the actual survival data from the Titanic.
Your program should process this information and compute statistics for the dataset. At the very minimum, you should display the number of survivors. Add more code, to answer questions that you find interesting –for example, comparing survival rates for adults vs. children etc.

Hints:

- Note that you need to obtain the data in the correct format, so that it works with the Passenger constructor. Use `nextInt()` to obtain the status; nextBoolean() to obtain the values for child and survivor; use `next()` for the sex.
- Remember to use `word.equals("female")` instead of `word == "female"` to test Strings for equality.

UML class diagram

Draw a UML class diagram that shows all your classes and their dependencies.