

Lab Exercise #4  
*Iterators and Serialization*  
Computer Science 2334

Group #: \_\_\_\_\_ Section #: \_\_\_\_\_  
Members: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

*Learning Objectives:*

- To reinforce the use of iterators.
- To learn how to use Serialization to write and read objects to and from files.
- To demonstrate this knowledge by completing a series of exercises.

*Instructions:*

This lab exercise requires a laptop with an Internet connection. Once you have completed the exercises in this document, your group will submit it for grading. All group members should legibly write their names at the top of this lab handout.

***Make sure you read this handout and look at all of the source code posted on the class website for this lab exercise before you begin working.***

Iterators will be an instrumental part of the projects you will create for this course. You must be familiar and comfortable using both Iterators and ListIterators. Serialization is an important feature of Java, that will be used in a future project. You must be able to read and write data to a file using Serialization.

For this lab exercise you will download and extend the solution for Lab Exercise #3. This code is posted on the class website. Download this code and carefully inspect how it works and the documentation comments included in the code.

1. Download the Movie.java and Test.java files from the class website. You will modify these files as a part of this lab exercise and submit them when you are finished. But, before you start modifying these files, first answer Question #2.
2. ObjectOutputStream and ObjectOutputStream can be used to read and write objects from and to streams. Combined with FileReader and FileWriter we can use these classes to read and write objects from and to binary files. What interface must the class, whose objects we want to read/write from/to files, implement?

3. Add the interface you chose for the answer of Question #3 to the declaration of the Movie class. For example, the declaration should read:

```
public class Movie implements Comparable, <Question #3 answer> {
```

4. Add a method with the following signature to the Test class that writes each Movie object that is stored in the movies list to a file, whose name is passed in as an argument to the method.

```
public static void writeEachMovie( String fileName, List movies )
```

The format of the file must be, where there are  $n$  movies:

```
<number of movies (n) as an Integer>
<movie #1>
<movie #2>
<movie #3>
.
.
.
<movie #n>
```

The code for this method will be similar to:

```
FileOutputStream fos = new FileOutputStream(<filename>);
ObjectOutputStream oos = new ObjectOutputStream(fos);
int n = <number of objects to write>
oos.writeInt( n );
while( <insert code here> ) // *** Use an Iterator for this loop ***
{
    oos.writeObject(<object to write>);
}
oos.close();
```

5. Add a method with the following signature to the Test class that reads in a list of movies from the file that the method writeEachMovie() creates.

```
public static List readEachMovie( String fileName )
```

The code for this method will be similar to:

```
FileInputStream fis = new FileInputStream(<filename>);
ObjectInputStream ois = new ObjectInputStream(fis);

int n = ois.readInt();
List newMovies = new ArrayList();
for( <from 0 to n> )
{
    Movie movie = (Movie) ois.readObject();

    <add movie to the list newMovies>
}
ois.close();
return newMovies;
```

6. Add code to the main method of the Test class that uses the methods `writeEachMovie()` and `readEachMovie()` to write and read the list of movies to a binary file. The code should follow the algorithm given below. Once you have written this code, test your program to ensure it writes and reads the binary file of movies.
  1. Write out the list of movies to a file.
  2. Erase all the elements in the list of movies.
  3. Print the contents of the list of movies to the console using `System.out.println()` and an Iterator.
  4. Read in the list of movies from the file in Step #1.
  5. Print the contents of the list of movies to the console using `System.out.println()` and an Iterator.

7. Add a new method to the Test class that has the signature given below. This method will write the entire list of movies to an output file using `ObjectOutputStream`.

```
public static void writeAllMovies( String fileName, List movies )
```

The method call to `ObjectOutputStream` should be similar to:

```
oos.writeObject(<list of movies>);
```

8. Add a new method to the Test class that has the signature given below. This method will read a complete list of movies from an input file using `ObjectInputStream`.

```
public static List readAllMovies( String fileName )
```

The method call to `ObjectInputStream` should be similar to:

```
<list of movies> = (List)ois.readObject();
```

9. Add code to the main method of the Test class that uses the methods `writeAllMovies()` and `readAllMovies()` to write and read the list of movies to a binary file. The code should follow the algorithm given below. Once you have written this code, test your program to ensure it writes and reads the binary file of movies.
  1. Write out the list of movies to a file.
  2. Erase all the elements in the list of movies.
  3. Print the contents of the list of movies to the console using `System.out.println()` and an Iterator.
  4. Read in the list of movies from the file in step #1.
  5. Print the contents of the list of movies to the console using `System.out.println()` and an Iterator.

10. Submit this lab electronically by transferring the `Movie.java` and `Test.java` files to `codd.cs.ou.edu` and save them in a folder named `lab4`. Then ssh into `codd.cs.ou.edu` and use the following commands to submit your lab.

```
cd lab4
```

```
/opt/cs2334/bin/submit cs2334-<section #> lab4 <*.java>
```

where `<section #>` is replaced by the section number for your lab section (011, 012, or 013) and `<*.java>` is either a list of the Java files to be submitted for the lab or the text `*.java`.

11. Turn in this lab handout to your lab instructor.