Lab Exercise #4
Serialization
Computer Science 2334

Section #:	
Members:	

Learning Objectives:

- 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.

Serialization is an important feature of Java, one that could 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 modify a project similar to Lab Exercise #3. Carefully inspect how it works and the documentation comments included in the code.

1. Download the Eclipse project archive from the class website. Import the project into your Eclipse workspace using the slides from Lab #2. You will submit the modified project when you are finished. But, before you start modifying these files, first answer Question #2.

2. ObjectInputStream 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 Word class, whose objects we want to read and write, implement?

3. Add the interface you chose for the above question to the declaration of the Word class. For example, the declaration should read:

```
public class Word implements <insert interface here> {
```

Also, do not forget to add your import statement for this declaration. If you are unsure of what package to import, then consult the online API, or your text.

4. Add a method with the following signature to the Word class that writes a Word object (in other words, a dictionary entry) to a file, whose name is passed in as an argument to the method. This method will be called from the Driver class.

```
public static void writeWord( String fileName, Word word )
```

The code for this method will be similar to:

```
FileOutputStream fos = new FileOutputStream(fileName);
ObjectOutputStream oos = new ObjectOutputStream(fos);
oos.writeObject(word);
oos.close();
```

Additionally, for this and all other "write" methods used in this lab, you must add:

```
throws IOException
```

to the end of each method signature.

5. Add a method with the following signature to the Word class that reads in a Word object from the file. Again, this method will be called from the Driver class.

```
public static Word readWord( String fileName )
```

The code for this method will be:

```
FileInputStream fis = new FileInputStream(fileName);
ObjectInputStream ois = new ObjectInputStream(fis);
Word word = (Word) ois.readObject();
ois.close();
return word;
```

Additionally, for this and all other "read" methods used in this lab, you must add:

```
throws IOException, ClassNotFoundException
```

to the end of each method signature.

6. Add code to the main method of the Driver class that uses the methods writeWord() and readWord() to write and read a Word object to/from 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.

a. Write out the Word object to a file.
b. Read in the Word object from a file.
c. Print the Word object to the console using System.out.println().

7. Add a new method to the Word class that has the signature given below. This method will write the entire list of dictionary entries (also called word objects) to an output file using ObjectOutputStream. Again, it will be called from the Driver class.

```
public static void writeDictionary( String fileName, List words )
```

The method call to ObjectOutputStream (inside the writeDictionary method) should be:

```
oos.writeObject(words);
```

8. Add a new method to the Word class that has the signature given below. This method will read a complete list of dictionary entries from an input file using ObjectInputStream. Surely, you know where it will be called from by now...

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

The method call to ObjectInputStream should be similar to:

```
List words = (List) ois.readObject();
```

9. Add code to the main method of the Driver class that uses the methods writeDictionary() and readDictionary() to write and read the list of dictionary entries to/from 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 list of items.

a. Create a list of Word objects. See Lab #3 for how to do this. Don't forget that you need to import a certain package to use the list.
b. Write out the list of dictionary entries to a file.
c. Erase all the elements in the list.
d. Print the list, which should be empty, to the console using System.out.println().
e. Read in the list of dictionary entries from the file used in step (b).
f. Print the contents of the list of dictionary entries to the console using System.out.println().

10. Submit the project archive following the steps given in the **Submission Instructions** by **9:00pm.**

11. Turn in this lab handout to your lab instructor. Please submit just one copy for your group.