CECS 277 LAB DEPENDENCY INVERSION

OBJECTIVE: Get some practice writing to an interface rather than a concrete implementation.

INTRODUCTION: Please remember the coding standards here.

In a way similar to what we did in lecture today with the geometric shapes, we are going to have you implement to an interface rather than concrete classes. You will start with the code here, and refactor it so that the StringWriterImp and StringReaderImp classes each implements an interface, and your StringProcessorDependent class binds to those interfaces, rather than concrete implementations. Finally, you have to redo DependencyInversionDependent so that it, too, creates variables that use those interface variable types rather than the concrete implementations.

PROCEDURE:

1. Build StringReader and StringWriter interfaces calls for everything but the constructors in StringReaderImp and StringWriterImp.
2. Update StringReaderImp and StringWriterImp so that they implement their respective interfaces instead of being independent classes. Be sure to put in the @Override annotations as appropriate.
3. Copy StringProcessorDependent and call the copy StringProcessor and change it so that it relies on your new interfaces rather than any specific implementation of those interfaces.
4. Copy DependencyInversionDependentRunner to DependencyInversionRunner and make that work with the new interfaces as well. Remember that you cannot create instances of an interface, but you can declare variables of an interface type.
5. The demo code that we went over in lecture is here.

WHAT TO TURN IN:

- StringReader.java
- StringWriter.java
- StringReaderImp.java
- StringWriterImp.java
- StringProcessor.java
- DependencyInversionRunner.java
- Your sample output, named console.txt