CS-162 Lab #4: Abstract classes & Interfaces

General

At the top of the every document that you create (word processing files or source files) include:

```
/**
 * Description of the class or document.
 *
 * @author YOUR NAME
 * @version CS162 Lab #, mm/dd/yyyy (replace with the last edit date)
*/
```

Submit in this lab via Moodle using the “Lab #” link. Your assignment must be uploaded by the assigned due date and time. Moodle will automatically close the link at that time. It is strongly highly recommend you do not wait until the very last minute to submit your work.

Concepts

This lab will extend your experience with designing and programming using Java Inheritance and add to your syntax knowledge the concepts of “**abstract classes**” and “**java interfaces**”.

Background

Read chapter #10 and review the online tutorials on inheritance. Study the example in chapter #10 that discusses the implementation of the Fox and Rabbit Simulation. This is an open ended lab where you can experiment with abstract classes and java interfaces. Just make sure that your final code meets the requirements listed in the Assignment section.

Assignment

For lab #4, please start with a completed version of Lab #2. You will be refactoring the code to explore both abstract classes and java interfaces.

Lab Requirements:

1. Copy one of the paired programmers Lab #2 project and name it Lab4.
2. Change the top level class in your project (the superclass) to be defined as an “abstract class”.
3. Write at least 2 new “abstract methods” for this class.
4. Implement the abstract methods (so that they are no longer abstract) in the subclasses. These methods should make use of some of the instance variables of the subclasses that are extensions of the superclass.
5. Create a java “interface” (see section 10.6 in your text) that declares two methods. They can be something like “isAlive( )” or “incrementAge( )”. Come up with 2 that make sense for one or two of your subclasses.
6. Have one or two of your subclasses also implement the interface that you defined in #5, in addition to the inheritance that it already has.
7. Write a main method in the appropriate class that creates the necessary object(s) and demonstrates the print method with at least 5 items of each subclass in your list.

Additional lab requirements:

1. Add JavaDoc comments for ALL public METHODS and CLASSES that you write
2. Add a JUnit test class for ONE of your classes, and create at least 2 JUnit test methods that test some aspect of your class.

Submission:

Submit your entire BlueJ project (zip only, no tar, 7zip, or other file compression types) folder for this lab into the CS162, Lab2 upload link (just as was done in CS161).