CECS 277 LAB INHERITANCE

OBJECTIVE: Give you your first hands-on exposure to inheritance, and what its limitations are.

INTRODUCTION: Please remember the coding standards here.

The Java Application Programming Interface (API) comprises a vast array (maybe even an armada) of packages and classes. As you mature as a developer, you will learn how to search for code that has already been tested and published, and make good use of that in the building of your own, custom, code. One way to make use of existing classes is to extend them to exhibit some specific behavior that you need that was not provided for in the original class. This strikes a nice compromise between building it from scratch, and suffering with a solution that does not really meet your needs.

PROCEDURE: The Java API documentation is here. Once you load that web page, select "FRAMES" from the menu bar at the top. Then select "java.desktop" in the upper left frame on that page, then select "Rectangle" from the classes in the lower left frame. This will bring you to the documentation for the java.awt.Rectangle class. We are going to extend that class and give it some new features in this lab.

If you comply with version 12 of the Java standard, you will have to put your classes into a `module`. That `module` creates an interface to the outside world: what it **exports** (exposes if you will and what it **requires** in order to run in terms of other modules. In this case, the java.awt Package is part of the java.desktop module (which the API documentation tells you at the very top). You will need to capture that information in your module-info.java file for your project.

1. Create a new Java project and call it "CECS 277 Lab Inheritance".
2. The wizard will prompt you for the name of the module that contains this package. Tell it "cecs277LabInheritance" to conform with the module naming conventions.
3. Inside that **project**, build a **package**, and call it "cecs277LabInheritance". The package groups one or more classes, and the module groups one or more packages.
4. Create a new class: `BetterRectangle` that `extends` the `java.awt.Rectangle` class. `BetterRectangle` will:
    a. Implement its **own** constructor with 4 integers as the arguments: x, y, width, and height. This constructor will simply rely on the constructor of the generic parent.
    b. Implement the getPerimeter () method to return the perimeter of the `BetterRectangle` as an `int`.
    c. Implement the getArea () method to return the square area of the `BetterRectangle` as an `int`.
    d. Override the toString method from the parent to output the string "Better Rectangle:" followed by the measurements of the `BetterRecangle` instance.

   i. Remember, those measurements are the x and y coordinates of the upper left hand corner of the BetterRectangle, and the width and height of the overall Rectangle.
   ii. Use the `toString` from **`Rectangle`** (the generic parent) in your `toString` in `BetterRectangle`.

5. Your driver program: InheritanceRunner starts with the code found here. Download that to drive your BetterRectangle class.
   a. For each section in InheritanceRunner, uncomment the code, and try to run the result.
   b. Document whether or not it compiles as comments in the code.
   c. If it does not compile, put comments into the code indicating **why** it did not compile.
   d. Then comment out that section again (if it will not compile) and go on to the next section.
   e. Be sure to include code that demonstrates your toString() method on BetterRectangle.

WHAT TO TURN IN:

- module-info.java
- InheritanceRunner.java
  - Be sure that your comments in the code reflect what you learned about each of those four sections in the code.
  - Comment out whatever you have to in order to get it to run.
- BetterRectangle.java
- Your console output, saved as a file called console.txt