MIST 4600. Spring 2014 Assignment D
Dr. Janine E. Aronson. Due: Wednesday, 11:00 p.m.

As always, your initial file is LastnameFirstname-AssignmentD. Download the zipped BlueJ Project start folder, unzip it, and rename the unzipped folder with your Lastname and Firstname. When you finish this assignment, put your name and a comment about what works and what does not at the top of the ClockDisplay class where indicated. Save the BlueJ Project, close the BlueJ Project, zip the folder properly and submit the zipped file properly to eLC Assignment D. Note that if after you submit your file, you may submit an update until the due date. eLC will retain your most recent submission. Note that if the testers do not work, then you earn 0. Your code must interface properly with the testers.

As always, I suggest that before you touch the keyboard, you look at and understand the existing code, and think about the task and how to accomplish it. Also, as soon as you make one or two changes to the code, recompile it.

Scenario: Do ALL your work in the ClockDisplayPlus class. This is a slightly modified clock-display BlueJ project that prints the time to the terminal window whenever time is changed (in method updateDisplay). You essentially must modify the ClockDisplay to include seconds, so that timeTick adds one second to the clock, not one minute. The display should always show hours:minutes:seconds. Obviously you need to make sure things roll over appropriately, from seconds to minutes, and from minutes to hours. One additional aspect is that before you modify the timeTick method, copy it into a method called timeTickMinute so that it is still possible to increment the clock by minutes if desired (for example, to reset the time like many digital clocks do).

Details:
You need to understand how the original clock-display works, and you will want to examine every single method and constructor and decide what (if anything) needs to be changed in each.

Before you begin, set the Terminal window for Unlimited buffering in order for me to test it properly.

Note, the steps below are outlined for success. This is the easiest path to completion. Such detailed steps may not be given in future Assignments.

Here is what you must do step-by-step:

a. The Wrapper and the Constructors:
Obviously, you need a new instance variable (attribute, field, global variable) called second that will be a numberDisplay. Add this in the wrapper with the rest of the instance variable declarations/definitions.

b. The Two Existing Constructors (yes, there are two):

In the constructor that has no parameters **and** the constructor into which you feed hour and minute, you will need to create the seconds numberDisplay field/attribute/instance variable/global variable (just like the hours and minutes numberDisplays), and set its initial value to the int 0. There are no additional parameters to these two constructors. Compile your code and verify that it works exactly as before. It should.

c. The New Constructor (yes, there will now be three):
You must write a new constructor that allows the user to enter a new int parameter second. Copy the code for the constructor into which you feed hour and minute into a new one. In this new one, add a third int parameter second. In this constructor, when you create the seconds NumberDisplay, feed this value into it. (Look in the tester classes to see how this constructor is used.) Before you can test this method, you must complete step d.

d. The setTime Method:
Modify the setTime method to include a third parameter int second, and use it to set the time properly. Make sure you compile and test the new constructor now.

e. The timeTick and timeTickMinute methods:
First, copy method timeTick and rename the copied method timeTickMinute. This way you can tick either by a minute or by a second. Method timeTick will increment by one second; method timeTickMinute will increment by one minute. (You could write a timeTickHour method as well for better functionality.)

Modify the timeTick method to increment the ClockDisplay by one second. This is a bit tricky. Exercise caution.

f. The updateDisplay method:
Modify this method to incorporate seconds when time is displayed.

Postscript:
There are tester classes that you should use when you have the new class written, but note that they will not work until you have created the new, third constructor.

Note that the testers use while loops – these create the repetition programming structure (e.g., look up the structured programming theorem). Loops are coming up in our next topic.

Don't forget to rename your project properly, and zip it properly, and put your name and code status in a comment at the top of the main code. Don't do one or more of these, then you will lose points.

When you finish all this, look back at all you have done. This Assignment involved juggling quite a few different aspects, variables, constructors and methods to make it work properly.