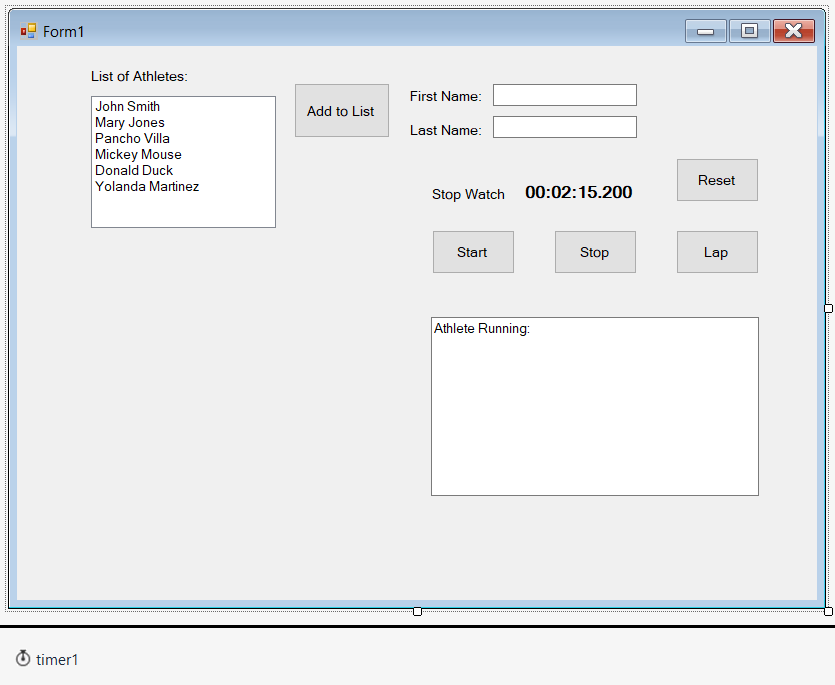
Use the any of the Time2 classes from previous assignments and create a new class that inherits Time2 (you can call it PreciseTime). This new class will add a field to keep milliseconds, so a PreciseTime object will have Hours, Minutes, Seconds (inherited) and milliSeconds as its data.

Create an Athlete class which will have (at a minimum) A First Name, a Last Name, and a List or an array (called say Laps) of PreciseTime objects as its data. Each PreciseTime object in the List/array will hold the time the athlete took to run one lap, so Laps[0] will have the time the first lap was completed, Laps[1] will be the time the second lap was completed, etc.

Create a Windows Application to keep track of Running Athletes performance while running laps. Something like (but not necessarily equal to) this prototype:



The functionality of the GUI should be as follows:

List of Athletes may be a listbox object (you can add Athlete’s objects to that list, not just string like the example in the book).

“Add to List” Button on the right should take the data from Fist and Last Name textboxes and create a new Athlete object (with Laps array set to zeros) and add him/her to the listbox. After entering all the athlete’s names (and creating all the athlete objetcs and putting them in the listbox), the user (coach) can start timing the athletes (new athletes can be added at any time, of course).

When an athlete’s name is selected in the listbox, that athlete’s data should be displayed in the Athlete’s running TextBox (lower right) and updated as information is entered in that athlete’s Laps List/Array (see below)

The Timer1 object should be set to a 1 millisecond resolution (maybe 10, to keep the resolution at 1/100th of a second-your choice). That it is should produce an event every millisecond (or every 10 milliseconds). Every event should update the StopWatch information accordingly. (StopWatch could also be a PreciseTime object).

Reset Button should set the StopWatch to 0:0:0.000

Start Button should enable the timer1 object to start producing periodic event every millisecond (or 10 milliseconds depending on your resolution), after each event the StopWatch should be advanced/incremented appropriately.

Lap Button should take the current value of the StopWatch and save it in the current lap being measured (first time in Laps[0], next click on Laps[1], etc.) for the athlete currently running.

Stop Button should disable the timer1 object so it won’t keep producing periodic events. It should store the last value in the StopWatch into the current/next Laps List/array entry.

EXTRA CREDIT:

(5 Points) Add an extra button that will report all athlete’s times in a new Window.

(15 Points) Make the report in the new Window sorted by the first lap time. You will need the Athlete’s class (and possibly the PriceiseTime class too) to implement the IComparable interface (CompareTo method) to be able to use the Sort method of the Array/List class.

(20 Points) Allow the coach to select the lap number whose time will be used to sort the report or also to sort the report by name. This could potentially involve the use of delegates. The compareTo method could use/call a delegate that points to the method that actually compares the objects and this delegate could be set to point to different methods that compare based on different criteria according to the coach’s choice when requesting the report. **This is challenging/NOT trivial, but very interesting**.