CS 205 - Programming for the Sciences  
Spring 2008 - Programming Assignment 1  
20 points

Out: January 17, 2008  
Due: January 24, 2008

Assignment

The roots of a quadratic equation of the form \( ax^2 + bx + c \) where \( a, b, \) and \( c \) are integer coefficients may be computed using the following equations:

\[
\frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad \frac{-b - \sqrt{b^2 - 4ac}}{2a}
\]

The number and type of the roots depends on the value of \( b^2 - 4ac \), called the discriminant. There are three possibilities:

<table>
<thead>
<tr>
<th>Discriminant</th>
<th>Nature of Roots</th>
</tr>
</thead>
<tbody>
<tr>
<td>( &gt; 0 )</td>
<td>Two real roots, one from each equation</td>
</tr>
<tr>
<td>( = 0 )</td>
<td>One real root, both equations give same root</td>
</tr>
<tr>
<td>( &lt; 0 )</td>
<td>Two complex roots, one from each equation</td>
</tr>
</tbody>
</table>

Write a C# Windows Application program that will accept the coefficients \( a, b, \) and \( c \) in textboxes, and display the number and type of the roots of the resulting quadratic equation and the values of those roots when a button is clicked. For the complex roots, display the value in the form \( x + yi \) where \( x \) is the real part \((-b)\), \( y \) is the magnitude of the imaginary part \((4ac - b^2)\), and \( i \) represents the square root of \(-1\).

Programming notes and an example GUI showing how this might look is given below.

What to submit

For full credit, a compressed (zipped) solution folder containing a C# project must be submitted as an attachment to an email to Dr. Hwang (hwang@evansville.edu) no later than 4:30pm on the due date.

To create a compressed solution folder find the solution folder via Windows Explorer. Right-click on the solution folder, select Send To, then select Compressed (zipped) folder. This will created a compressed folder of the same name as the solution folder with extension .zip and an icon of a folder with a zipper.

Programming notes:

- The square root method is named `Math.Sqrt`. It takes one argument and returns the square root of the argument, e.g. `Math.Sqrt(4)` returns 2.
- The input textboxes, the button, and result box must be renamed to descriptive names
- There must be enough labeling for a user to understand what she is expected to do.
- The form must have an appropriate titlebar.
- Error checking is not required on the input. (However, ill-formatted input will cause the program to crash.)

Typos fixed 01/17/08
GUI notes
The GUI interface of the program might look like the following for each possible result:

- The coefficient textboxes have the TextAlign property set to Right.
- The result area is a textbox with the ReadOnly property set to true and the Multiline property set to true.
- The ability to use TAB to move from control to control is configured using the TabStop and TabIndex properties. By default, the TabStop property is set to true for all input controls. The first created control is given TabIndex value of 0 and the TabIndex is automatically incremented for each subsequently created control. When the application is started, the input control with the lowest TabIndex value is given focus. Pressing TAB then moves focus to the input control with the next lowest TabIndex value, etc. If the input controls were not created in TAB order, change the TabIndex for each input control to the correct order.