Assignment #6

CIT1156 – Programming II

Objective:
Developing skills implementing C# syntax, including:

-GUI events
-event handlers
-control properties
-creation of a custom class
-defining class attributes
-define class properties

Overview:

Create a graphical application that will allow the user to enter some information on the players of a hockey team. This information will only include the player's name, jersey number and the number of goals scored. These details will be stored in an array (no more than 30 players) that is created using the custom class HockeyPlayer.

Steps:

1. Define the custom class of HockeyPlayer that will be used to store the data for each player entered into this application. The attributes/properties that are required are player's name, jersey number and goals scored. Create the **player's name** as a **read-only attribute** that has a property that can retrieve the player's name. Use the auto-implemented property to define read-only access for the jersey number. The goals scored property can be implement in any manner that you would like but must allow updates.

2. Define an array of HockeyPlayers as an attribute of the form. In the form's constructor create this array that will hold at a maximum of 30 players.

3. Design the form to have the appropriate number of text boxes for the information that is required for one player. Please label the text boxes so the user will know how to enter the data.

4. Place a button on the form that is labeled as "Add Player". Once the user has completed the entry of the all information this button's event will create an object of type HockeyPlayer using the information that is entered and add the object to the array that holds all players.

5. Once the user has all the players entered it would be nice to get a list of them. Add another button and a richTexBox to the form. Have this new buttons event display the information for each player on one line in nicely aligned columns.

   Optional:

6. A sorted display would be really nice. If you choose, implement the IComparable interface and required method (see page 404 of the text book). If you had an attribute of the class that controlled which attribute was used to compare the objects you could allow the user to specify the sort order (ie Alphabetically, by jersey number, goals).