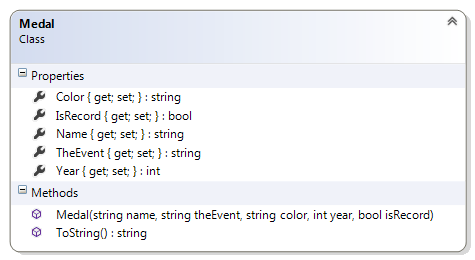
# The Medal Class

### Code the Pet class below:

This class comprise of five properties, a constructor and a ToString() method. All the properties have public read access and private write access.



#### Description of class members

##### Properties:

**Color** – this is a string representing the color of this object (Of course the only colors should be Gold, Silver and Bronze, but the only this can be achieved to is by using Enums). The getter is public and the setter is private

**IsRecord** – this is a bool indicating if this was a record setting event. The getter is public and the setter is private

**Name** – this is a string representing the holder of this object. The getter is public and the setter is private

**TheEvent** – this is a string representing the event of this object. (Event is a reserved word in C#). The getter is public and the setter is private

**Year** – this is an integer representing the year of this object. The getter is public and the setter is private

##### Methods

**Medal(string name, string theEvent, string color, int year, bool isRecord)** – This public constructor takes five arguments: a string representing the name, a string representing the event, a string representing the type of medal, an integer representing the year and a bool indicating if a World Record or Olympic Record was set in this event. It assigns the arguments to the appropriate fields.

**ToString()** – This public method overrides the ToString of the object class. It does not take any argument and returns a string representation of the object. You may return a string in the format “2012 - Boxing(R) Narendra(Gold)”.

If the event is not a record event then the “(R)” should not be present in the output. The ToString() method.is the best place to implement this feature.

### Test Harness

Insert the following code statements in the **Main()** method of your Program.cs file:

//create a medal object

Medal m1 = new Medal("Horace Gwynne", "Boxing", "Gold", 2012, true));

//print the object

Console.WriteLine(m1);

//print only the name of the medal holder

Console.WriteLine(m1.Name);

//assign a new object to m1

m1 = new Medal("Michael Phelps", "Swimming", "Gold", 2012, false));

//print the update m1

Console.WriteLine(m1);

//create a list to store the medal objects

List<Medal> medals = new List<Medal>();

medals.Add(new Medal("Ryan Cochrane", "Swimming", "Silver", 2012, false));

medals.Add(new Medal("Adam van Koeverden", "Canoeing", "Silver", 2012, false));

medals.Add(new Medal("Rosie MacLennan", "Gymnastics", "Gold", 2012, false));

medals.Add(new Medal("Christine Girard", "Weightlifting", "Bronze", 2012, false));

medals.Add(new Medal("Charles Hamelin", "Short Track", "Gold", 2014, true));

medals.Add(new Medal("Alexandre Bilodeau", "Freestyle skiing", "Gold", 2012, true));

medals.Add(new Medal("Jennifer Jones", "Curling", "Gold", 2014, false));

medals.Add(new Medal("Charle Cournoyer", "Short Track", "Bronze", 2014, false));

medals.Add(new Medal("Mark McMorris", "Snowboarding", "Bronze", 2014, false));

medals.Add(new Medal("Sidney Crosby ", "Ice Hockkey", "Gold", 2014, false));

medals.Add(new Medal("Brad Jacobs", "Curling", "Gold", 2014, false));

medals.Add(new Medal("Ryan Fry", "Curling", "Gold", 2014, false));

medals.Add(new Medal("Antoine Valois-Fortier", "Judo", "Bronze", 2012, false));

medals.Add(new Medal("Brent Hayden", "Swimming", "Bronze", 2012, false));

//print all the medals

//print only the names of all the medal holder

//print all the gold medals

//print all the medals in 2012

//print all the world record medals