**Submission: See course shell for how and when to submit**
You will practice using Properties and iterating a list.

## The MedalColor Enum

**Code the MedalColor enum below:**
This enum consist of 3 constants which does not require any special treatment (such as setting any Flags attribute).

| MedalColor |
|---|
| enum |
| **Constants** |
| Bronze<br>Silver<br>Gold |

## The Medal Class

**Code the Medal class below:**
This class comprise of five properties, a constructor and a `ToString()` method. All the properties are public readonly.

| Medal |
|---|
| Class |
| **Properties** |
| **+** «property setter absent» Name        : **string** |
| **+** «property setter absent» TheEvent    : **string** |
| **+** «property setter absent» Color       : **MedalColor** |
| **+** «property setter absent» Year        : **int** |
| **+** «property setter absent» IsRecord    : **bool** |
| **Methods** |
| **+** «constructor» Medal(<br>    name            : **string**,<br>    theEvent        : **string**,<br>    color           : **string**,<br>    year            : **int**, |

```
        isRecord            : bool)
 +   ToString()                              : string
```

## Description of class members

Properties:

All the properties have public getter and the setter is absent making them all readonly properties

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

**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 absent.

**Color** – this is an enum representing the color of this object The getter is public and the setter is absent.

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

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

Fields:

No fields are defined in this class

Constructor:

**public Medal(string name, string theEvent, MedalColor 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.

Methods:

**public override string 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:

```csharp
//create a medal object
Medal m1 = new Medal("Horace Gwynne", "Boxing", MedalColor.Gold, 2012, true);
//print the object
Console.WriteLine(m1);
//print only the name of the medal holder
Console.WriteLine(m1.Name);

//create another object
Medal m2 = new Medal("Michael Phelps", "Swimming", MedalColor.Gold, 2012, false);
//print the updated m2
Console.WriteLine(m2);
```

```csharp
//create a list to store the medal objects
List<Medal> medals = new List<Medal>(){ m1, m2};

medals.Add(new Medal("Ryan Cochrane", "Swimming", MedalColor.Silver, 2012, false));
medals.Add(new Medal("Adam van Koeverden", "Canoeing", MedalColor.Silver, 2012, false));
medals.Add(new Medal("Rosie MacLennan", "Gymnastics", MedalColor.Gold, 2012, false));
medals.Add(new Medal("Christine Girard", "Weightlifting", MedalColor.Bronze, 2012, false));
medals.Add(new Medal("Charles Hamelin", "Short Track", MedalColor.Gold, 2014, true));
medals.Add(new Medal("Alexandre Bilodeau", "Freestyle skiing", MedalColor.Gold, 2012, true));
medals.Add(new Medal("Jennifer Jones", "Curling", MedalColor.Gold, 2014, false));
medals.Add(new Medal("Charle Cournoyer", "Short Track", MedalColor.Bronze, 2014, false));
medals.Add(new Medal("Mark McMorris", "Snowboarding", MedalColor.Bronze, 2014, false));
medals.Add(new Medal("Sidney Crosby ", "Ice Hockey", MedalColor.Gold, 2014, false));
medals.Add(new Medal("Brad Jacobs", "Curling", MedalColor.Gold, 2014, false));
```

```
medals.Add(new Medal("Ryan Fry", "Curling", MedalColor.Gold, 2014, false));
medals.Add(new Medal("Antoine Valois-Fortier", "Judo", MedalColor.Bronze, 2012, false));
medals.Add(new Medal("Brent Hayden", "Swimming", MedalColor.Bronze, 2012, false));

//prints a numbered list of 16 medals.
Console.WriteLine("\n\nAll 16 medals");

//prints a numbered list of 16 names (ONLY)
Console.WriteLine("\n\nAll 16 names");

//prints a numbered list of 9 gold medals
Console.WriteLine("\n\nAll 9 gold medals");

//prints a numbered list of 9 medals in 2012
Console.WriteLine("\n\nAll 9 medals");

//prints a numbered list of 4 gold medals in 2012
Console.WriteLine("\n\nAll 4 gold medals");

//prints a numbered list of 3 world record medals
Console.WriteLine("\n\nAll 3 records");

 //saving all the medal to file Medals.txt
Console.WriteLine("\n\nSaving to file");
```