**.NET class and gcnew**

**C# version:**

|  |  |
| --- | --- |
| **Ex14-02CS.cs** | |
| **Line#** | **Code** |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | using System;  namespace Ex14\_02CS {  class Circle {  double r;  public Circle(double r) {  Radius = r;  }  public double Radius {  get { return r; }  set {  if (value < 0) throw new Exception("Radius can't be -ve!");  r = value;  }  }  public double Area {  get { return Math.PI \* r \* r; }  }  public double Circumference {  get { return 2 \* Math.PI \* r; }  }  }  class Program {  static void Main(string[] args) {  Console.WriteLine("Circle demo in C#");  Circle c = new Circle(100);  Console.WriteLine("Radius:{0:f2}",c.Radius);  Console.WriteLine("Area:{0:f2}",c.Area);  Console.WriteLine("Circumference:{0:f2}", c.Circumference);  Console.ReadKey();  }  }  } |

**C++ CLR version:**

|  |  |
| --- | --- |
| **Ex14-02CPP.cpp** | |
| **Line#** | **Code** |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | #include "pch.h"  using namespace System;  ref class Circle {  private:  double r;  public:  Circle(double r) {  Radius = r;  }  property double Radius {  double get() { return r; }  void set(double value) {  if (value < 0) throw gcnew Exception("Radius can't be -ve!");  r = value;  }  }  property double Area {  double get() { return Math::PI \* r \* r; }  }  property double Circumference {  double get() { return 2 \* Math::PI \* r; }  }  };  int main(array<System::String ^> ^args){  Console::WriteLine("Circle demo in C++");  Circle^ c = gcnew Circle(100);  Console::WriteLine("Radius:{0:f2}", c->Radius);  Console::WriteLine("Area:{0:f2}", c->Area);  Console::WriteLine("Area:{0:f2}", c->Circumference);  Console::ReadKey();  return 0;  } |