**Delegates**

**C# version:**

|  |  |
| --- | --- |
| **Ex14-08CS.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  35 | using System;  namespace Ex14\_08CS {  class SomeClass {  public static void Func(String str) {  Console.WriteLine("static SomeClass::Func - {0}", str);  }  };  class OtherClass {  public OtherClass(Int32 n) {  num = n;  }  public void Method(String str) {  Console.WriteLine("OtherClass::Method - {0}, num = {1}",  str, num);  }  Int32 num;  };  delegate void MyCallback(String str);  class Program {  static void Main(string[] args) {  Console.WriteLine("Delegate in C#");  MyCallback callback = new MyCallback(SomeClass.Func);  callback("single");  callback += new MyCallback(SomeClass.Func);  OtherClass f = new OtherClass(99);  callback += new MyCallback(f.Method);  f = new OtherClass(100);  callback += new MyCallback(f.Method);  callback("chained");  Console.ReadKey();  }  }  } |

**C++ CLR version:**

|  |  |
| --- | --- |
| **Ex14-08CPP.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  36  37 | #include "pch.h"  using namespace System;  ref class SomeClass{  public:  static void Func(String^ str){  Console::WriteLine("static SomeClass::Func - {0}", str);  }  };  ref class OtherClass{  public:  OtherClass(Int32 n){  num = n;  }  void Method(String^ str){  Console::WriteLine("OtherClass::Method - {0}, num = {1}",  str, num);  }  Int32 num;  };  delegate void MyCallback(String^ str);  int main(){  Console::WriteLine("Delegate in C++");  MyCallback^ callback = gcnew MyCallback(SomeClass::Func);  callback("single");  callback += gcnew MyCallback(SomeClass::Func);  OtherClass^ f = gcnew OtherClass(99);  callback += gcnew MyCallback(f, &OtherClass::Method);  f = gcnew OtherClass(100);  callback += gcnew MyCallback(f, &OtherClass::Method);  callback("chained");  Console::ReadKey();  return 0;  } |