

```
1 using System;
2 using System.Collections.Generic;
3
4 namespace exam1
5 {
6     class MyClass
7     {
8         public string InstanceMethod(int a, int b)
9         {
10             return (a + b * 2).ToString();
11         }
12     }
13     class Program
14     {
15
16         public delegate string MyDelegate(int arg1, int arg2);
17
18
19         //----- the two functions that match the delgate signature -----
20         public static string Function1(int a, int b)
21         {
22             return (a + b).ToString();
23         }
24
25         public static string Function2(int a, int b)
26         {
27             return (a * b).ToString();
28         }
29
30         public static void Main(string[] args)
31         {
32             //-----indexer-----
33             List<int> myList = new List<int>();
34             myList.Add(100);
35             myList.Add(200);
36             myList.Add(300);
37             myList.Add(400);
38
39             Console.WriteLine(RandomClass.Randomize(myList));
40
41             //-----delegates-----
42
43             //-----normal delegate-----
44             MyDelegate f1 = Function1;
45             Console.WriteLine("the number from the first function is " + f1
46                             (10, 20));
47
48             MyDelegate f2 = Function2;
49
50             Console.WriteLine("the number form the second function is " + f2
51                             (10, 20));
```

```

51
52     // we can ever chain the delegate like so
53     MyDelegate f1f2 = f1 + f2;
54     Console.WriteLine("the number of the chained delegate is" + f1f2 (10, 20));
55     f1f2 -= f1;
56
57     //-----delegate using a mthoed from another class-----
58
59     MyClass mc = new MyClass();
60     MyDelegate f3 = mc.InstanceMethod;
61     Console.WriteLine("the number is " + f3(10, 20));
62
63
64     //-----2anonymous delegate-----
65     MyDelegate f4 = delegate (int arg1, int arg2)
66     {
67         return (arg1 + arg2).ToString();
68     };
69     Console.WriteLine("the number is " + f4(10, 20));
70
71     //-----predicate delegate -----
72     Predicate<string> isUpperCase = upperCase;
73     bool doesItMeetTheCrieteriea = isUpperCase("HELLO");
74     Console.WriteLine(doesItMeetTheCrieteriea);
75
76     // -----extension-----
77
78     Time time1 = new Time();
79     time1.Hours = 2;
80     time1.Minutes = 3;
81     Time time2 = new Time();
82
83     time2.Hours = 4;
84     time2.Minutes = 5;
85
86     Time time3 = time1 + time2;
87     Console.WriteLine(time3.Hours);
88     Console.WriteLine(time3.Minutes);
89
90     Console.ReadLine();
91 }
92
93 private static bool upperCase(string something)
94 {
95     return something.Equals(something.ToUpper());
96 }
97 }
98 }

```