

Lambda Expression

Lambda Expression

- Is an anonymous function
- Use to create delegates
- Use to create expression tree
- Write local function
- Passed as argument of function call
- Use as return value of function call

Exercise : 00100 Assign Lambda to delegate

Create anonymous function with Lambda expression

```
// anonymous function without param
d myD = () => 12;
int k = myD();
Console.WriteLine(k);

// anonymous function 1 param
Square mySquare = x => x * x; // assign expression to delegate type
int j = mySquare(5); // call delegate x = 5 j = 25
Console.WriteLine(j);

// anonymous function 2 params
Sum mySum = (x, y) => x + y;
int temp = mySum(2, 4);
Console.WriteLine(temp);
```

Exercise 00110 : Using Lambda expression in WHERE

Method-based query by using the Enumerable.Where standard query operator

```
// Data source.  
int[] scores = { 90, 71, 82, 93, 75, 82 };  
  
// The call to Count forces iteration of the source  
int highScoreCount = scores.Where(n => n > 80).Count();  
  
Console.WriteLine("{0} scores are greater than 80", highScoreCount);
```

Exercise 00120 : Lambda and LINQ SUM

Use lambda expression in LINQ SUM method to get group sum value

```
var categories =  
    from student in students  
    group student by student.Year into studentGroup  
    select new  
    {  
        GradeLevel = studentGroup.Key,  
        TotalScore = studentGroup.Sum(s => s.ExamScores.Sum())  
    };
```

Exercise 00130 : Lambda and Event

Use a lambda expression to define an event handler form click

