Lambda Expressions / Anonymous Functions

named function - a **function with a name** that can **invoked from anywhere in the code** accept parameters and return a value.

named function are used when you want to call the function multiple times from different place in your code.

Named function that takes 2 parameters and returns their sum:

```
int sum(int n1, int n2) {
                        return n1 + n2;
                      }
```

Named function that takes no parameters and returns a 0:

```
int giveMeAZero() {
                    return 0;
                  }
```

Lambda Expression - a **function without a name** that can **only be invoked where it is defined** it can accept parameters and return a value
(aka arrow function or anonymous methods/functions)

Lambda Expressions are used where a function is needed in only one placed in the code.
(Usually as parameters to other methods)

instead of *function* name you code **=>** after the parameters and before the **{**

Named function that takes 2 parameters and returns their sum:

```
int sum(int n1, int n2) {
                          return n1 + n2;
                        }
```

Lambda Expression that takes 2 parameters and returns their sum:

```
//  int sum(int n1, int n2) {

    (int n1, int n2) => {
                         return n1 + n2;
                         }

    (int n1, int n2)  => { return n1 + n2; }  // single-line processes may be coded
                                              //  on one line

    (int n1, int n2)  =>  return n1 + n2;     // C# does not {} if a single line process
```

Lambda Expression that takes no parameters and returns a 0:

```
//  int giveMeAZero() {

    () => {
            return 0;
          }

    () => { return 0; }
```

Lambda Expressions are frequently used as arguments for other methods.

Some methods require function be passed to them that they then execute as part of thier processing. The functions that are passed are referred to a **callback functions**.

When a function is used as a parameter to another function, we refer to the parameter function as a **callback function**.

The method receiving the **callback function**, calls the callback function as part of it's processing.

Callback methods may either named or Lambda Expressions.