Anonymous Delegates

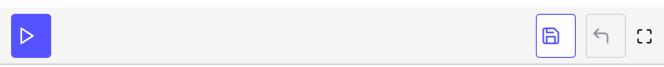
his lesson briefs over anonymous delegates in C#.

Anonymous delegates are a short way to write delegate code, specified using the delegate keyword. The delegate code can also reference local variables of the function in which they are declared.

Anonymous delegates are automatically converted into methods by the compiler.

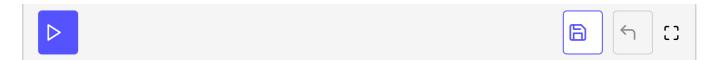
For example:

```
using System;
delegate void Procedure();
class DelegateDemo2
    static Procedure someProcs = null;
    private static void AddProc()
        int variable = 100;
        someProcs += new Procedure(delegate
                Console.WriteLine(variable);
            });
    }
    static void Main()
        someProcs += new Procedure(delegate { Console.WriteLine("test"); });
        AddProc();
        someProcs();
    }
}
```



They can accept arguments just as normal methods can:

using System; delegate void Procedure(string text); class DelegateDemo3 static Procedure someProcs = null; private static void AddProc() int variable = 100; someProcs += new Procedure(delegate(string text) Console.WriteLine(text + ", " + variable.ToString()); }); } static void Main() someProcs += new Procedure(delegate(string text) { Console.WriteLine(text); }); AddProc(); someProcs("testing"); } }



Let us now learn about what multicasting is and how it's done in C#!