C#: Delegates, Events, and Lambdas

# Delegates

Delegates are placeholders for functions, variables for functions.

Delegates can be dynamically chained together.

A delegate signature is the datatype of the value that is returned.

Delegates are type safe. The compiler checks the function signatures of delegates when you use them at runtime. This helps prevent common errors, like passing wrong arguments of the wrong type or getting the wrong datatype of results back.

Delegates can be dynamically switched at runtime.

Using references as inputs allows a method to change the value of a passed variable.

# Events

Events are used to broadcast and listen to messages.

Events are based on delegates and how they communicate asynchronously.

Events can be switched on and off as needed.

There are standard events, but users can create custom events.

The .NET framework defines a predefined event handler as a template that you can use to send events using the same format that .NET does.

Events can be chained together.

# Lambdas

Lambdas are functionally the same as regular delegates but are written using a more concise syntax that can be easier to read.

Lambdas are another way of writing anonymous functions.

Expression lambda

Statement lambda