Delegates are a pipeline that link the event raiser to the event handler. They define the return types and param the event handler must deal with. The delegate signature has to be mimicked by the handler.

Hook up an event raised to event handler:

1. Instantiate an instance of a delegate with a ref to the event handler
2. Add it to the even raiser with the += sign.

When using the EventHandler built in delegate for an event pipeline, the corresponding Event handler must have params Object, EventArgs.

Delegate inference – instead of instantiating a delegate instance to attach to the event, you simply state the event handler method name as the input to the += for the event. Behind the scenes the compiler generates the delegate instance

+= tab tab will generate the delegate inference and the non-implemented method stub.

Delegate inference will also work with -=.

Anonymous method – a method without a name. In this context they allow event handlers to be hooked directly to the event using syntax similar to object initialisation. Disadvantage of anonymous methods is they can’t be used anywhere else. These can be used for simple processing. Its similar to a lambda.