Research five advanced Visual Studio Debugger features that you have never used before in your development. Describe the capabilities and purpose for each of the five features. For two of the features, demonstrate the feature using one of the in-class activities, and provide a screenshot showing its usage.

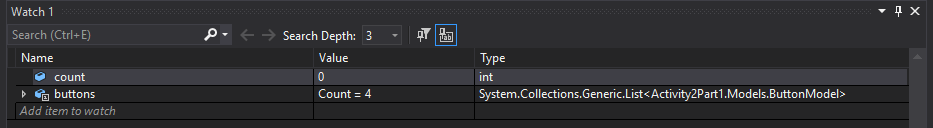
**Run to Cursor**

Hot Key: Ctrl+F10

While editing code you can right-click within the editor and select “Run to Cursor” which will set a temporary breakpoint without setting a hard breakpoint that will iterate each time the debugger is launched. The debugger will stop at the first breakpoint it hits, whether this is your Run to Cursor, or a hard breakpoint so be cautious where you are using this.

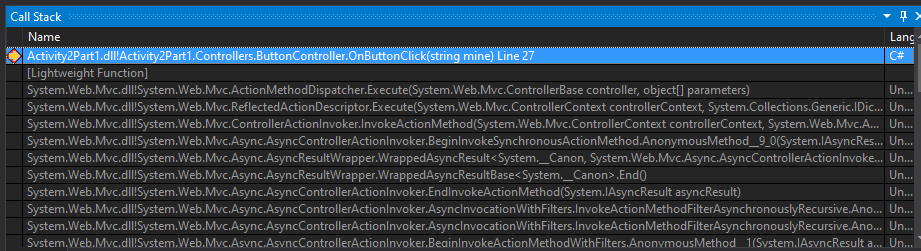
**Set a Watch Window**

Using a watch window can help step through variables and expressions. Simply right click the object and select “Add Watch” to add an abject to a watch window. To Open a watch window, select Debug > Windows > Watch > Watch1 or Hot Key: Ctrl+Alt+W > 1 (2, 3, or 4 work as well if you have multiple watch windows). Step into or F11 will advance through variables and expressions while within the watch window.



**Examine a Call Stack**

Similar to the watch window a Call-stack window can show and iterate through a call stack while showing the value representations of each as methods and functions are call and completed. Other stack frames, functions, methods and even a different thread can also be reached from within the call stack window.



**Set Function Breakpoints**

Hot Key: Alt+F9 > Ctrl+B

Debug > New Breakpoint > Function Breakpoint

Breaking execution when a function is called can be very helpful to ensure that the correct function is indeed being called (overloaded options), and can also help to locate the function in large programs when the name is known, but not the location.

**Track an Object ID with a Conditional Breakpoint**

Create Object IDs for specific instances of reference types. These Object IDs can the be used by breakpoint conditions within a watch window. The Object ID is generated by the common language runtime (CLR) debugging services and associated with the object.

In the **Locals** window, right-click the desired variable and select “**Make Object ID” to create the Object Id which can then be traced**.

Jones, M., Kramer, J., & Hogenson, G. (2019, February 13). Debugging in Visual Studio - Visual Studio. Retrieved February 03, 2021, from https://docs.microsoft.com/en-us/visualstudio/debugger/?view=vs-2019