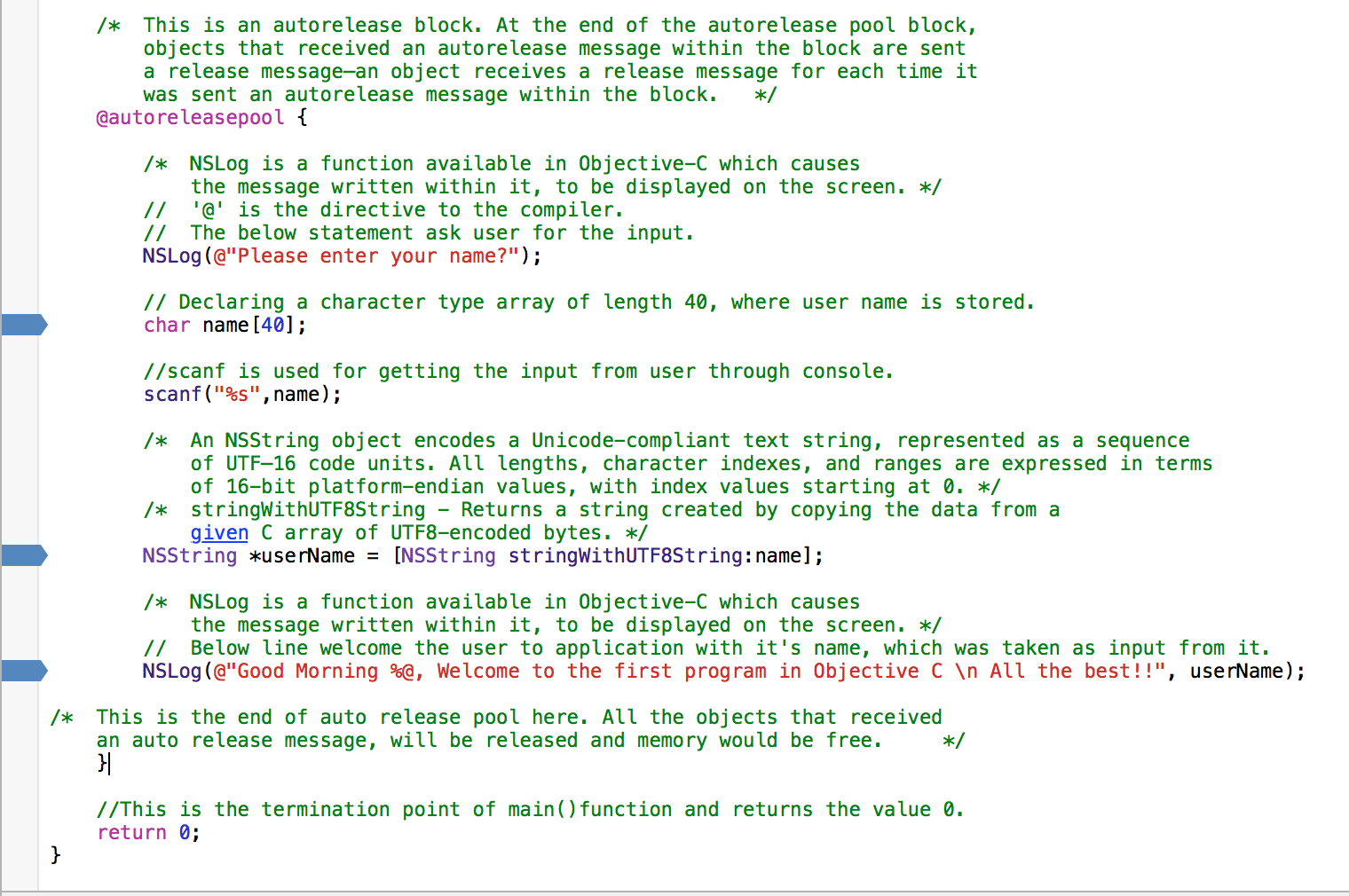
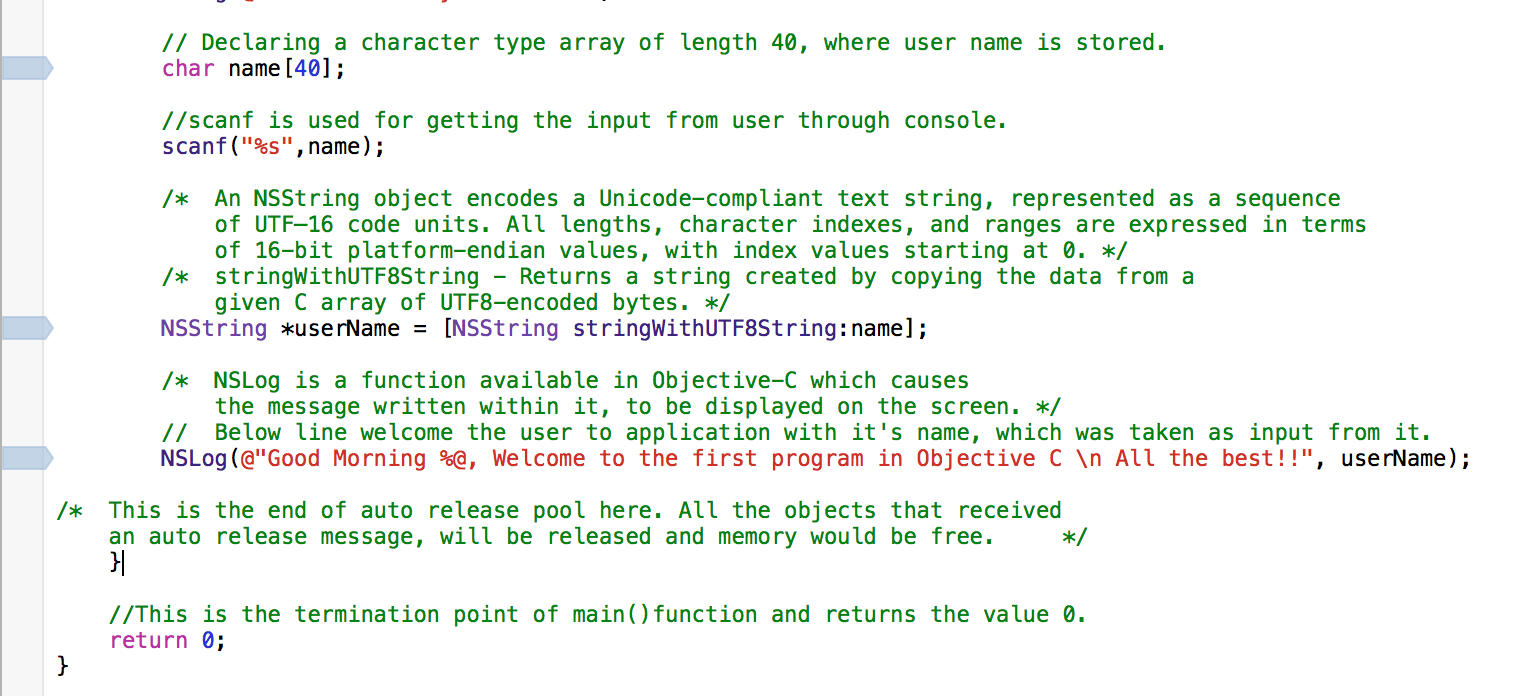
**DEBUGGING ENVIORNMENT**

We can add breakpoints at any time, before or after your code is already running. There’s no need to recompile your app for breakpoints to work. To add a breakpoint, click the line number next to the code where you would like to pause. A blue arrow will appear indicating the breakpoint’s position:



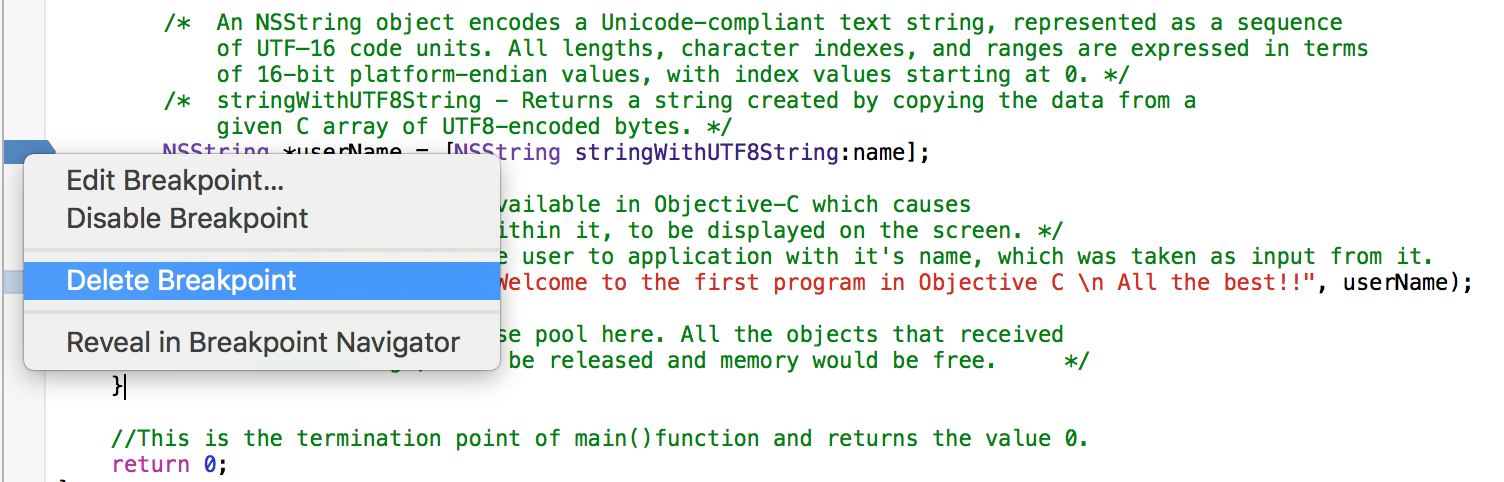
This is a simple breakpoint. When you run your app and it reaches this point in execution, everything will pause until you tell the process to continue on.

We can temporarily disable the breakpoint by clicking it again. The breakpoint will become transparent to indicate it’s temporarily disabled.

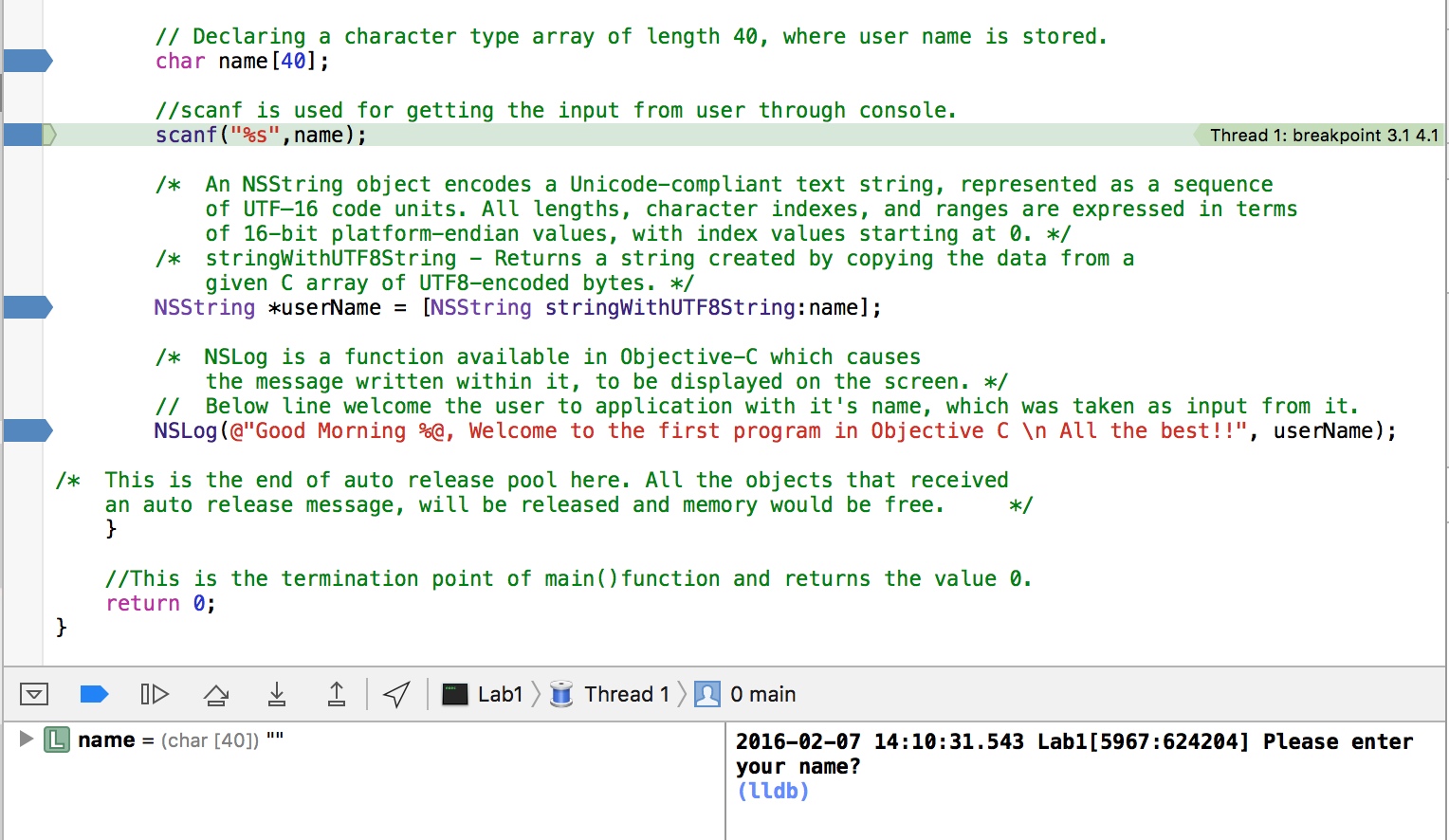


We can simply click it again to re-enable it. Breakpoints you disable will be ignored and act as if they aren’t there.

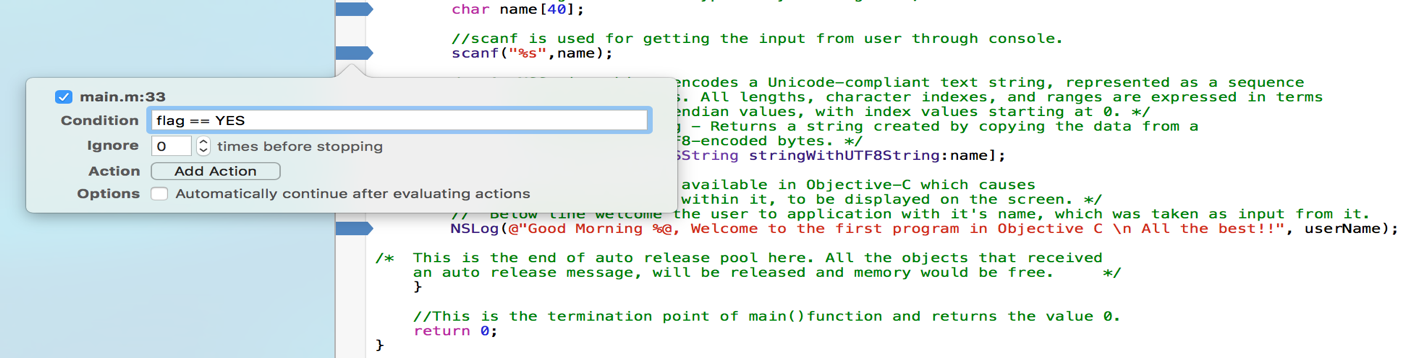
To remove a breakpoint, you either right-click and select “Delete Breakpoint”, or drag the breakpoint off the bar.



The breakpoint stops **before the current line is executed** so if you want to see the next log message, you need to *continue* execution using the debug controls:

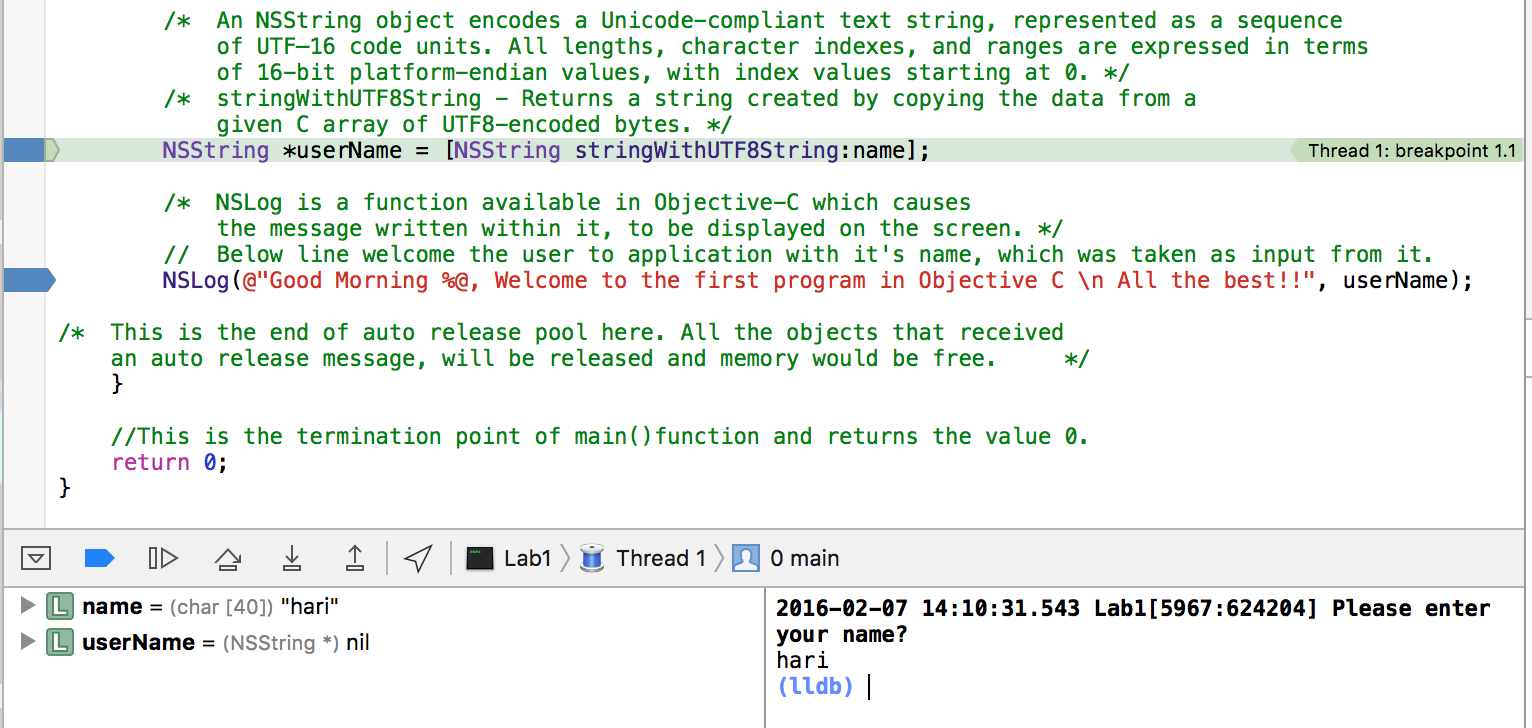


The below is the conditional Breakpoint, these breakpoints will execute when the value of Boolean variable flag value is true. If flag is false, it won’t run.



**Step Over:**

Step Over will directly go to the next line of code, without stopping on any other user code that might be called in the current line. F6 is the keyboard shortcut



**Step In**

Step into will jump either to the next line or to the first line of any user function method that is executed by the current line. We use step in when we want to jump into a method that is being called by the current line of the code.

**Step Out**

Step out executes the rest of the current method or function.

