**Summary of Debugging in Xcode**

There are two different Errors when we code:

* **Compilation Errors:** These errors and bugs show up when your app is compiled (or “built”). For example, you’ve made a typo and the Swift compiler chokes on it.
* **Runtime Errors:** These errors and bugs happen when your app is running on an iPhone. The way your app works leads to an error or exception, and your app crashes.

Compilation errors typically show up in the Xcode editors, and usually in the Console. Runtime errors always show up in the Console, and occasionally in the Xcode editor.

A few common bugs in Swift are:

Off-by-one error (or Index out of bounds)

SIGABRT

Use of Unresolved Identifier

Unrecognized Selector Sent to Instance

Why to find a bug in the code:

A great approach to finding and fixing bugs in Xcode is by using breakpoints. With a breakpoint, the Xcode debugger can stop your code at almost any moment and show you the exact state of your app at that point. You can do that by adding a breakpoint to a line of code in your app.

When the app executes, and then “hits” that line of code, execution of your code comes to a halt, and you can look inside the guts of your app. You can inspect the exact values of variables, properties etc. at that point. And from there, you can step line-by-line through your code.

In Xcode we have some tools to use with breakpoints:

1. Hide the debug area
2. Deactivate breakpoints
3. Continue program execution
4. Step over
5. Step into
6. Step out

More:

In some cases, you can’t set a breakpoint in your code, because a bug occurs in a framework or Cocoa Touch library that you don’t control. When you can’t reach those lines of code, you can’t set a breakpoint. You might know them from Swift error handling with do-try-catch. Some functions can “throw” exceptions when an error occurs.