Debugging

Basics

There are three types of bugs:

Compile Time Errors

These types of bugs prevent your code from being compiled. They are mostly caused by syntax errors in the code. These are the easiest to resolve; just fix the syntax. These are usually caused by: * Missing Semicolons. * Misspelled class names. * Misspelled methods. * Not importing the class that has those methods or objects.

Run-Time Errors

These are the type of bugs that only appear when you run your code. You may know them as exceptions. But they don't have to exceptions, they can be caused by passing in the wrong type of variable (A string instead of a char). Breakpoints are a great aid in figuring out what part of the code caused the run-time error. You can even define your own exceptions.

Logical Errors

This type of error won't always show up in the debug console. It's when code runs, but it doesn't do what it's suppoused to do. These are probably because of logical errors in expressions in the code. There is no one perfect way to pinpoint the main cause. The general strategy is to use print statements or logs to examine the values. Check each method for logical soundness, make sure that it does what it is supposed to do on a bunch of cases. Make sure to include border cases.

Using the LLVM Debugger

The current version of XCode uses the LLVM debugger, pervious versions used to use GDB. Using a debugger, and taking the time to learn how to use one will speed up the debugging process. I highly recommend to take a look at apple's guide to using the LLVM debugger.

You can use the debugger to execute your app one instruction at a time.

Image from: http://www.peterfriese.de/
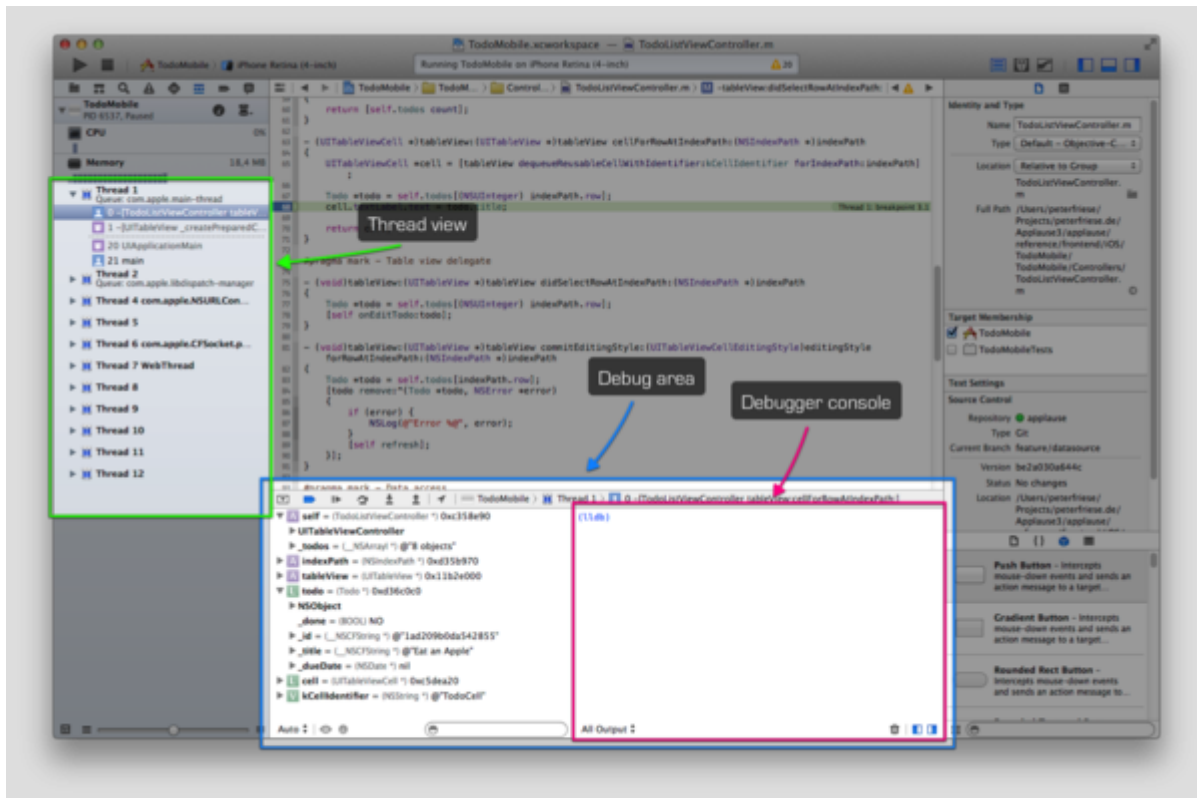
These are some useful commands you can use:

Printing objects

`po objectName` or `print object objectName`

Printing Primitives

`p variableName` or `px variable name`, if you want to print the hex value.

Display Variable Values Every Time Execution is Stopped

`display variableName`

Display All Global and Static Variables in the Current Source File

`target variable` or `ta v`

Display Contents of Global or Static Variable

`target variable variableName` or `ta v variableName`

Additional Links:

- Apple's LLDB Quick Start Guide.
- LLDB Debugger Tutorial.
- Raywenderlich's "My App Crashed, now what?" Part1, Part2.