Debugging
## Debugging Differences Between:

<table>
<thead>
<tr>
<th>Xcode 3</th>
<th>Xcode 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Separate debugger window</td>
<td>Integrated debugger</td>
</tr>
<tr>
<td>Separate-ish GuardMalloc</td>
<td>Integrated-ish GuardMalloc</td>
</tr>
<tr>
<td>Separate-ish Zombies</td>
<td>Integrated-ish Zombies</td>
</tr>
<tr>
<td>Run and Run &amp; Debug</td>
<td>Run, breakpoints on/off</td>
</tr>
<tr>
<td>Separate Tools (e.g. Shark)</td>
<td>“Instruments”</td>
</tr>
<tr>
<td></td>
<td>Static analyzer</td>
</tr>
</tbody>
</table>
Some Debugging Friends

• `assert(expression);`

• `NSLog(@"The object = %@", anObject);`

• `GuardMalloc`

• Initial breakpoints, single stepping, etc.

• Watching variables

• Testing, testing, and testing (to follow).
assert(expression)

• Raises a signal if expression is false
• Disabled by #define-ing NDEBUG
  • Common to see within #ifndef DEBUG
• Should contain zero functionality (e.g. ‘=’)
• Very common to use for checking inputs
  • Can also (sanity) check outputs too
NSLog()

• NSLog(NSString* format, ...);
• NSLog(@"The object = %@", anObject);
  • Calls [anObject description]
  • Which is <classname:address> by default.
• Override -(NSString*)description to {essentially} create your own “toString()”
GuardMalloc

- Aligned to 16-byte boundaries, so it may not catch (smaller) buffer overruns.
- Can write your own “sentinel” code if you need finer-grained checking.
Breakpoints

• Helpful to stop/break and step through code when it is first written.

• Breakpoints can also be conditional, or have attached actions.

• Click on them to temporarily disable them. Drag them off of the window to delete them.
Watching Variables

- Right-click variable in debugger and select “Watch variable…”
- Any time the value changes, you will drop into the debugger at the line of code that changed that variable’s value.
Hardcore

- You can change variables' values in the debugger while the program is running.
- You can change the PC when you are paused in the debugger.
Resources

- Debugging and Analyzing Your Code (Apple)
- iOS Debugging (Apple)