Project #0: Assembly Language with IAR Embedded Workbench

**The Tutorial:**

Our impression of the tutorial was that of bewilderment. The problem is that it didn’t really help give clarity in how to write programs in the IDE. The basic introduction to the IDE and the basic Assembly language introductions given in the help file were much more informative. Even with the built in introduction files and the paper tutorial, the basic structure of what an assembly program in the IDE should look like (header files, naming external files, etc.).

**The Assembly Program:**

The assembly program was fairly simple. The hardest part was setting the program up for thumb instructions as well as understanding the basic vector table. We used the stack window to demonstrate the output of the program. The stack trace and the source code for this program are shown in Appendix A, the debug log and the build log are shown in Appendix B. The disassembly of the program is shown in Appendix C.

**The C++ Program:**

The C++ program was also simple. The biggest advantage to the C++ program was the ability to exclude the preamble necessary at the beginning of the assembly file. The source code and the output of this program are shown in Appendix D, the debug log and the build logs are shown in Appendix E, and the disassembly of this program is shown in Appendix F.

**The Comparison:**

Again, with the exception of the preamble in the assembly code, the source code of these two programs is very similar. As shown by Appendix A and D, there are about 8 effective lines of code in each source. The size comparison of these two programs is astounding. The project folder, including the workspace of the assembly language program is about 136 kb and the C++ program is about 1.23mb! That means the C++ program is 9 times the size of the assembly language program. The C++ program was significantly simpler to program on the basis that there was no setup involved other than including the iostream. If you compare the disassembly code as shown in Appendix C and F, they are really similar…at least in the main section of the code. If you notice the assembly code in Appendix C, preceding and following the main section of the code is just a bunch of operations that deal with zeros. This leads me to believe that this code serves no real purpose. Now if you observe the code in Appendix F, the main section is nearly the same, but the preceding and following code actually appears to be functional. This also doesn’t mention the branches that are going off in every direction in the C++ code. This leads us to the conclusion that the assembly code as opposed to the C++ code is much more efficient and executes at a much faster rate.

**The Conclusion:**

If the solution to a programming problem is time sensitive and speed/storage space is not very crucial, then C++ is the way to go. If storage space is a problem and speed is crucial, assembly language is by far the most effective solution.

**The Contributions:**

Zac wrote the C++ program, Andrew wrote the assembly program and assembled the report, Brandon was a late joiner to the group and got up to speed on IAR programming.