Reading for This Lecture

• Norm Matloff’s Debugging Tutorial
Compilers

- GCC
- Intel
- Portland Group
- Borland
- Microsoft Visual Studio
- Oracle Studio
- xIC
Integrated Development Environments

- Microsoft Visual Studio
- Eclipse
- Anjuta
- Dev-C++
Editors

• IDEs
• Emacs
• Notepad++
Other Tools

- Version Control (CVS/SVN)
- Make
- CMake
- Autotools
Debugging

- IDEs
- GDB
- DDD
- Valgrind
- Electric Fence
- Purify
Profiling

- Gprof
- Quantify
Good Development Practices

• Use version control
• Make code readable
• Formatting
• Comments and documentation
• Naming conventions
• Develop good unit tests
• Make code reusable
• Modularity
• API
Debugging

- Debugging is a process of verifying that certain invariants that you expect actually hold.

- When the code is not working, you must look for inconsistencies that indicate a violated invariant.

- Modularity and good unit testing make this process much easier.

- Example: Debugging Insertion Sort
Memory debuggers

- Memory debuggers are tools that can help you find errors such as reads/writes to unallocated memory.
- They can also help you find memory leaks.
- These types of errors are particularly difficult to find in practice because they may not cause a crash.
- They also may cause random behavior that could be different from one run to the next.
Profilers

- A profiler can help you determine where the bottlenecks are in your code.
- The profiler will tell you
- How many times each function was called
- How much time was spent in each function