CS2141 – Software Development using C/C++

C++ Background

Origins of C++

- Initially C++ was built on top of C
- C was developed in the early 1970's by Kernighan and Ritchie at Bell Labs
- C++ was developed by Stroustrup in the 1980s.
- Initially C++ was a set of macros and library routines for C
- The new C++ was explicitly intended to be backward compatible with C

The Legacy Problem

- Basing C++ on C has caused some problems:
  - Use of libraries that predate C++ (such as stdio)
  - Use of the preprocessor to create constants, rather than `const` variables
  - Use of simple string functions that manipulate arrays of characters, rather than the newer `string` type
  - Inconsistent approaches to representing booleans
  - Overuse of global variables
- Usually these problems will only be encountered in older code

Philosophy of C and C++

- Programmers should not have to pay for the features they do not use
- As a result of this philosophy:
  - There are very few run-time checks
  - There are about a half dozen cases where the outcome of an operation is left to the platform
    - Examples - integer division or mod with negative operands
    - These tend to be things that are normally not used

Comparison with Java

- Java gave priority to simplicity and security instead of emphasizing efficiency
- Therefore:
  - Choices left to the programmer in C++ are eliminated
  - Java eliminates all situations left unspecified by C++
  - Java's memory model is simplified by always using the slower heap-based model
  - Java provides much stricter run-time checks

C++ vs. Java

- Java was intended for embedded systems such as VCRs where programs must work correctly and securely, even if this means running slower
- C++ was intended for systems programming - writing small programs that execute quickly and efficiently
- The different goals make it hard to say that one language is “better” than the other