C++: Architecture and Conventions

Plan for today

- A first look at C++
  - Today: Nuts and Bolts
  - Monday: Data types, operation, statement, and pointers
  - Tuesday: Functions and scope, arrays, basic I/O

Before we start

- Announcement
  - The Academic Accommodation Office is looking for readers to read CS4 text into a digital recorder.
  - Pay is $6.33 / hr
  - Interested? Contact ASAP
    - Learning Development Center (01-2310)
    - 475-2023
    - aaoldc@rit.edu

Before we start

- Any questions on UML from last week?

History of C++

- 1969 – UNIX
  - Created to play space wars
  - B (Basic Combined Programming Language)
- 1970
  - UNIX port to PDP-11
- 1972
  - C (rewrite of B)
- Early 1980s
  - C++ invented as a "superset of C"

History of C++

- Why bring this up
  - C is a "low level" system language
    - Program as manipulator of memory
    - All memory management done by hand
  - C++ is not only based on C but is a proper superset
    - If you can do it in C you can do it in C++
    - Added Object Oriented paradigm.
  - Take home message: C++ is not Java!!!!

File Structure

- One class, two files
  - Header file (.h)
    - Contains class declaration (interface++)
  - Source file (.c, .cpp, .C, .cxx)
    - Contains class definition
      - implementation of methods

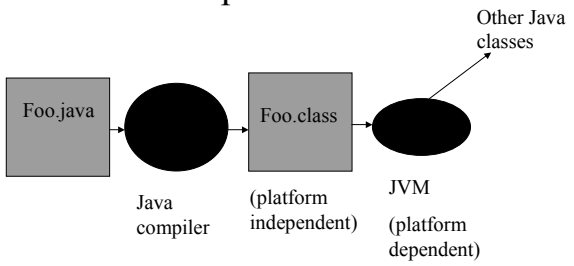
File Structure

- Cpp – The C Preprocessor
  - Reads all C code before compilation
  - Directives
    - Including text files
      - #include
    - Conditional compilation
      - #if / #ifdef / #ifndef / #else / #endif
    - Macros
      - #define

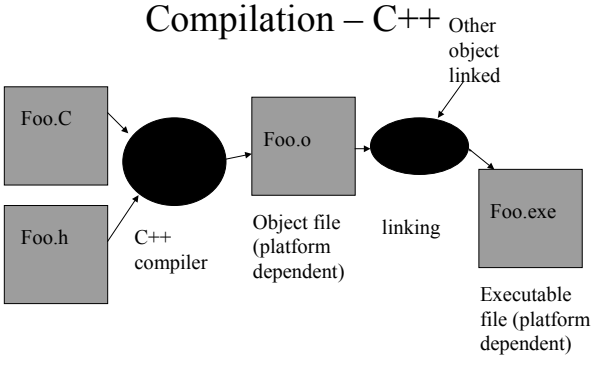
File Structure

- #include "filename.h"
  - Inserts text from one file into another before compilation
  - Contain info needed by other files to compile
    - Libraries – function signature
    - Classes – class interface (I.e. header file)

Compilation – Java



Compilation – C++



Compilation – Java vs C++

- Java
  - Compilation unit is the class
  - External classes located using a predefined path
  - JVM needed to execute
- C++
  - Compilation unit is the file
  - External classes require header and precompiled object file
  - Executable file need to execute.

Compilation – C++

- Use CC to compile individual files into object files
- Use CC to link object files into an executable file
  - Need to specify executable name, otherwise will be named a.out
- Run the executable file.

Using C Libraries

- Like Java, C/C++ has a multitude of useful auxillary functions and classes in libraries
- Unlike Java, C++ does not have the notion of packages.
- C / C++ also doesn't have nice javadocs
- Use `man` instead

Using C Libraries

- Library header files must be #included for compilation
  - #include "mylib.h"
  - #include <math.h>
- Library object files must be linked when linking.
  - CC file file file –lm mylib.o

Managing C++ Projects

- `make`
  - Files to be compiled
  - Compiler options
  - Libraries
  - Dependencies
  - Make will build an executable
  - `makemake` – Makes Makefiles!
  - Lab 3

Running C++ Executables

- No top level "main" class
  - main() in its own file
  - main (char *argv[], int argc)
- If there is a problem…
  - `Bus error (core dumped)`
  - `Segmentation fault (core dumped)`
- `Questions?`

Anatomy of a class

- Let's take a look at a header and source file for a C++ class, shall we?

Anatomy of a class

- Things to remember
  - .h & .C files
  - No package, but namespaces
  - C++ Style guidelines
  - Comment, comment, comment
- Questions?

Next time

- The 50 minute tour of C++.
- Any questions?
- Have a good weekend.